PROGRAMMAZIONE - STORIA - INTERVISTE - RECENSIONI - APPROFONDIMENTI

Future days are back

RETROMAGAZINE WORLD

NUMERO 49 - NOVEMBRE 2024 - WWW.RETROMAGAZINE.NET

ISSN 3103-697X

DARK KNIGHT

IL RITORNO DI BATMAN SU ZX

ATARI PONG

120 PAGINE!

Scansioni digitali = Memoria permanente

Sono da poco passate le 23:30. Domani dovrò svegliarmi presto per un impegno di lavoro già fissato da tempo. Sono già a letto, ma il sonno, dopo una giornata stressante passata davanti al PC, stenta a raggiungermi. Anzi, proprio non ne vuole sapere di prendermi. Alla fioca luce di una lampada, la stanza sembra assumere un colore ambra scuro. Con un gesto quasi automatico cerco sul comodino di fianco il tablet d'ordinanza e lo apro sull'app di accesso al NAS casalingo, una parte del quale è dedicato al nostalgico mondo del retrocomputing, un archivio colmo di articoli tecnici, manuali, libri e riviste pubblicati soprattutto negli anni 80 e 90. Seleziono la cartella "Sinclair" e poi "Mags" e poi apro il numero 24 di Sinclair User, edizione inglese del marzo 1984. Il sommario riporta vari articoli per lo ZX Spectrum ed una presentazione/recensione dell'allora nuovo gioiello della casa fondata da Sir Clive, il Sinclair QL. Dopo aver scorso qualche articolo di programmazione e lo speciale sul QL, finalmente le braccia accoglienti di Morfeo mi avvincono.

Il mattino dopo, mentre vado all'appuntamento di lavoro, ripenso alla "porta del tempo" che ho potuto aprire la notte precedente, verso un mondo che non c'è più, ma a cui posso accedere ogni volta che voglio. E mi son chiesto: chi devo ringraziare per questa fantastica possibilità? Ovviamente la casa editrice che in quegli anni pubblicava la rivista, ma anche il lavoro di un singolo appassionato, di un gruppo organizzato o dell'intera comunità dedita al retrocomputing che ha effettuato la digitalizzazione. Di certo non scopro io oggi l'importanza dell'attività di scansione digitale. Ma qui mi preme soprattutto sottolineare alcuni punti chiave che evidenziano la grande utilità di questi sforzi compiuti da gruppi di agguerriti appassionati.

Preservazione della conoscenza storica: la scansione digitale permette di conservare la conoscenza legata alla tecnologia del passato, documentando come si siano sviluppati hardware e software nel corso dei decenni. Questo patrimonio culturale e tecnico è infatti, dopo tanti anni, a rischio di scomparsa.

Accessibilità universale: con la digitalizzazione, chiunque, in qualunque parte del mondo, può accedere a materiali rari senza dover possedere fisicamente una copia cartacea.

Conservazione di documentazione tecnica: manuali e riviste spesso contengono informazioni tecniche cruciali, come schemi elettrici, specifiche hardware, procedure di programmazione, che consentono il restauro e la manutenzione di vecchi computer e console.

[continua e finisce in quarta di copertina...]

SOMMARIO

	PAG.
◊ Atari PONG	Pag. 3
◊ Atari 2600 – Tecnologie AI...	Pag. 9
◊ The SPECTRUM	Pag. 13
◊ Il Nintendo 64	Pag. 16
◊ Donkey Kong su PC128 Olivetti Prodest	Pag. 18
◊ Il C64 incontra l'Intelligenza Artificiale	Pag. 23
◊ Zena	Pag. 29
◊ Commdore 64 - BASIC V2	Pag. 32
◊ Inform 6 - Tutorial 4	Pag. 34
◊ Fino ad arrivare laddove nessun SID...	Pag. 43
◊ Il peso del Tetris	Pag. 52
◊ Giappone 25^puntata: Gachagacha...pon!	Pag. 58
◊ Fireteam Rogue per Super Nintendo...	Pag. 60
◊ Firenze Vintage Bit 2024	Pag. 64
◊ Quinty (SNes)	Pag. 67
◊ Escape 2042 (Jaguar)	Pag. 68
◊ Kudzu (GB)	Pag. 70
◊ Zephyr's Pass (GBC)	Pag. 72
◊ King's Quest VI (Atari ST)	Pag. 74
◊ Spelunky (C64)	Pag. 76
◊ Dark Knight (ZX)	Pag. 78
◊ Hayato's Journey (MD)	Pag. 80
◊ Super Dassalo Land (GBC)	Pag. 82
◊ Black & White (NEXT)	Pag. 84
◊ Data Man (NES)	Pag. 86
◊ Ruff in Trouble (Atari 8bit)	Pag. 88
◊ Super Mario Bros Mini (GBC)	Pag. 90
◊ Super Mario Eclipse (GameCube)	Pag. 92
◊ Tony Montezuma's Gold (C64/...)	Pag. 94
◊ Jester (NES/SNes)	Pag. 97
◊ Dr. Dangerous (Amiga)	Pag. 98
◊ Parasol Stars (Switch)	Pag. 100
◊ Portal Buster (C64)	Pag. 102
◊ Binary Battle (C64/...)	Pag. 104
◊ Shift (C64/...)	Pag. 106
◊ Metal Slider Glory... (Super Famicom)	Pag. 108
◊ Qyxar (Coleco)	Pag. 110
◊ Pokettohiro! (GBC)	Pag. 111
◊ Spanky's Quest (GB)	Pag. 112
◊ The Curse of Lies (MSX2/...)	Pag. 113
◊ Lunar Lancer (GB)	Pag. 114
◊ Tiburoncy (C64)	Pag. 115
◊ Aliens Neoplasma 2 (ZX/...)	Pag. 116
◊ Twist of Souls (Atari 8bit)	Pag. 117
◊ Chimney Man (MSX)	Pag. 118
◊ Back to the Future II (C64)	Pag. 119

Hanno collaborato alla stesura di questo numero di RetroMagazine World (in ordine sparso):

- Carlo N. Del Mar Pirazzini
- Daniele Brahimi
- Francesco Fiorentini
- Paolo Cattaneo
- Roberto Del Mar Pirazzini
- Giampaolo Moraschi
- Leonardo Miliani
- Takahiro Yoshioka
- Barbara "Morgana" Murgida
- Eugenio Rapella
- Javier Gonzalez
- Marta Rossmann
- Gianluca Girelli
- Marco Pistorio
- Ksenia Kamash
- Michele Ugolini
- Evan Granovitz
- Alberto Apostolo
- Ingrid Poggiali
- David La Monaca
- Bimbomillo
- Fredrik Ramsberg

- Immagine di copertina: **Giuseppe Mangini**
- Layout di copertina: **Carlo N. Del Mar Pirazzini**

Atari PONG

di Leonardo Miliani

Nel numero 41 della rivista, pubblicato circa 2 anni fa, abbiamo parlato della storia della VCS/2600, una console di Atari che ha segnato un'epoca trasformando i videogiochi in una vera e propria industria. Ma questa console non è la prima di successo né quella che ha lanciato Atari nell'Olimpo dei produttori di videogiochi. Prima di essa l'azienda fondata da Nolan Bushnell aveva rilasciato un altro prodotto di successo, che aveva infuso nell'azienda fama e soldi. Un prodotto talmente iconico che, nella seconda metà degli anni '70 del XX secolo, era stato clonato da praticamente qualunque produttore di dispositivi elettronici: stiamo parlando di PONG (fig. 1).

La storia che si ripete

Nolan Bushnell, durante le vacanze estive nel periodo del college, lavora presso un parco giochi e lì inizia a formarsi in lui la mentalità imprenditoriale osservando come le persone spendono soldi nei giochi elettromeccanici del parco. Successivamente frequenta il corso di ingegneria dell'Università dello Utah di Salt Lake City e qui entra in contatto con uno dei primi, rudimentali, videogiochi: è proprio presso questa università che è stato creato, pochi anni prima, Spacewar, un videogioco a tema spaziale dove due astronavi si danno battaglia. Il gioco gira su un PDP-1 e si comanda tramite rudimentali joystick. Dopo l'università si sposa e si trasferisce nella Silicon Valley, andando a lavorare presso la Ampex, che produce sistemi per la registrazione audio/video su nastri magnetici: qui conosce l'ingegnere elettronico Ted Dabney, con cui stringe amicizia. Successivamente i due vengono a conoscenza del fatto che è stata realizzata una versione ridotta di Spacewar basata su un PDP-11, chiamata Galaxy Game: rispetto all'originale, il computer e tutta l'elettronica sono contenuti in un unico cabinato e che, cosa più importante agli occhi di Bushnell, il gioco è azionato da una gettoniera. A Bushnell tornano in mente i soldi versati nelle gettoniere dei giochi del parco in cui lavorava da ragazzo e matura l'idea di realizzare un proprio videogioco. Con Dabney fondano la Syzygy nel 1971 e creano il loro primo arcade, "Computer Space". A differenza di Spaceware e Galaxy Game, Computer Space è molto più semplice, essendo basato su logiche TTL e non su un computer, per cui risulta anche più economic. Propongono il gioco alla Nutting Associates, che lo distribuisce senza successo: il gameplay

Fig. 1 - Il cabinato Atari PONG originale(foto: Patrick Nafarrete – fonte: Wikimedia Commons)

Fig. 2 - Alan Alcorn (foto: Alex Handy - fonte: en.wikipedia)

è difficile per delle persone non ancora abituate ai videogiochi e più portate alle classiche macchine da intrattenimento dell'epoca. Quando propongono di realizzarne una versione semplificata per risollevarne le vendite, Nutting Associates si tira indietro rifiutandosi di distribuirla.

(se volete approfondire la storia dei primi anni di Bushnell e della stessa Atari, vi consiglio di rileggere il su citato articolo sull'Atari VCS/2600 dal numero 41 della rivista)

Lo sviluppo di PONG

Bushnell e Dabney capiscono che le attuali società distributrici di giochi elettromeccanici non sono ancora pronte alla rivoluzione elettronica e decidono di distribuire in proprio i giochi che da quel momento produrranno. Hanno però bisogno di ulteriore aiuto e decidono di assumere un loro ex-collega, un altro ingegnere con cui hanno lavorato alla Ampex, Alan "Al" Alcorn (fig. 2). Bushnell e Dabney hanno in mente di realizzare un gioco di guida ma pensano che il compito sia troppo impegnativo per Alcorn, non solo perché è completamente a digiuno di videogiochi ma anche perché Alcorn è un ingegnere analogico mentre per i videogiochi che ha in mente Bushnell sono richieste conoscenze digitali. E qui entra in ballo la console Magnavox Odyssey. Nel mese di maggio del 1972, prima della presentazione ufficiale, Magnavox fa un tour attraverso diversi Stati degli USA per raccogliere le impressioni pre-lancio circa la console ed i suoi giochi. Ad una delle tappe partecipa anche Bushnell e la sua attenzione viene catturata da uno dei giochi che, nonostante i suoi difetti, pensa abbia del potenziale: il Tennis. Ci sono un paio di "racchette" che, dai lati dello schermo, fanno rimbalzare una "pallina" per lo schermo.

Bushnell decide di usare questo gioco come banco di prova per far prendere confidenza ad Alcorn con la realizzazione dei videogiochi. Per stimolare l'ingegnere, gli racconta una piccola bugia: gli dice che ha per le mani un contratto con la General Electric per un dispositivo che deve essere consegnato entro un certo lasso di tempo. Questo dispositivo ha 3 elementi fissi: deve esserci un oggetto mobile (una "pallina") che rimbalzi per lo schermo, deve avere 2 paddle per comandare altrettante "racchette"

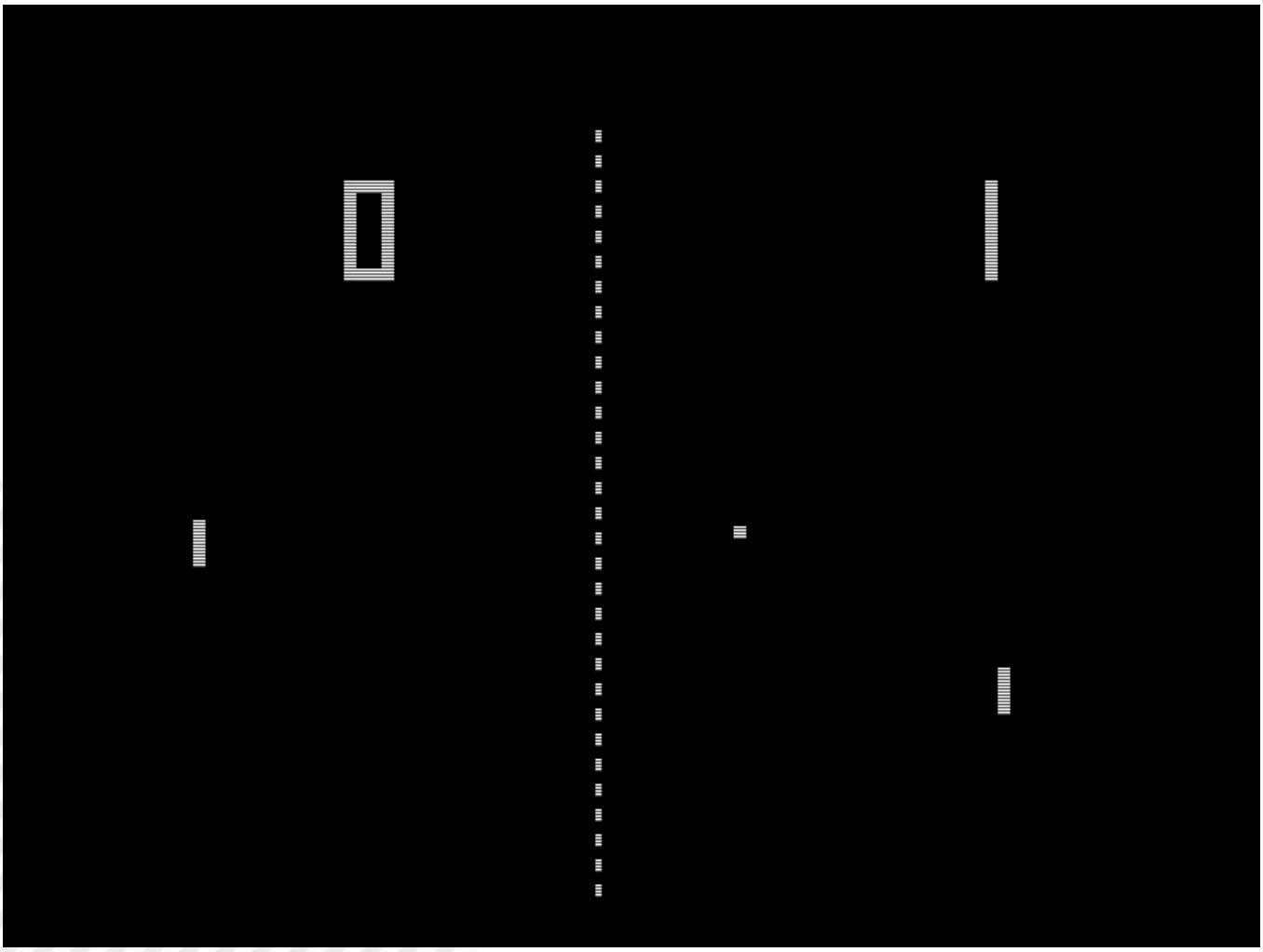


Fig. 3 - Lo schermo di gioco di PONG: 2 racchette fanno rimbalzare una pallina, un tabellone del punteggio in alto segna i punti dei 2 giocatori (foto: Bumm13 - fonte: En.wikipedia)

Fig. 4 - L'Andy Capp's Tavern, come appare oggi (un nuovo locale ha preso il suo posto nel 1984), dove nel 1973 è stato installato il prototipo del PONG (foto: CoolCaesar - fonte: Wikimedia Commons)

e deve mostrare un punteggio a video. Alcorn inizia a lavorare sul progetto e, nel giro di qualche mese, realizza un prototipo che risponde ai requisiti richiesti, aggiungendo anche alcune caratteristiche che fanno del suo videogioco qualcosa di molto più intrigante rispetto al Tennis dell'Odyssey. Ad esempio, la pallina rimbalza a diverse angolazioni: per ottenere ciò, Alcorn ha diviso il segmento della racchetta in 8 segmenti più piccoli e, a seconda di quello colpito, la pallina cambia angolo. Anche la velocità di quest'ultima non è sempre uguale: ad ogni rimbalzo aumenta, per aumentare il grado di difficoltà, tornando a quella di partenza solo nel caso in cui esca da uno dei due lati (quindi quando un giocatore "cicca" la pallina in una ribattuta). Il punteggio viene mostrato con dei numeri in alto simili a quelli mostrati dai display a LED a 7 segmenti: vince il primo dei giocatori che arriva ad 11 punti (fig. 3). Curiosamente, un problema del sistema di controllo delle racchette tramite paddle non viene corretto da Alcorn, ma lasciato come ulteriore difficoltà per il giocatore: nel sistema sviluppato dall'ingegnere, infatti, si manifesta un difetto per cui le racchette non possono raggiungere la parte più alta dello schermo, permettendo quindi alla pallina che viaggia molto in alto di non poter essere presa dal giocatore. Alcorn in seguito ha dichiarato che questo "difetto/caratteristica", la velocità della pallina che aumenta ad ogni colpo a segno e la differente angolazione del rimbalzo sono elementi presenti affinché esista un sistema per impedire che 2 bravi giocatori continuino una partita all'infinito.

Alla fine dello sviluppo, Bushnell si accorge che il gioco ha del potenziale. Chiede perciò ad Alcorn di aggiungere anche del sonoro: il problema che deve affrontare Alcorn è serio perché il sistema è realizzato interamente con componenti digitali e non è facile ottenere dei segnali analogici senza incrementare la complessità della scheda fino ad allora realizzata. Alla fine risolve il problema utilizzando il circuito di generazione del segnale video, ottenendo dei suoni utilizzando i componenti elettronici già presenti.

Alla fine di tutto questo lavoro, quel gioco che doveva essere una semplice palestra per far fare pratica ad Alcorn è risultato essere un videogioco molto interessante.

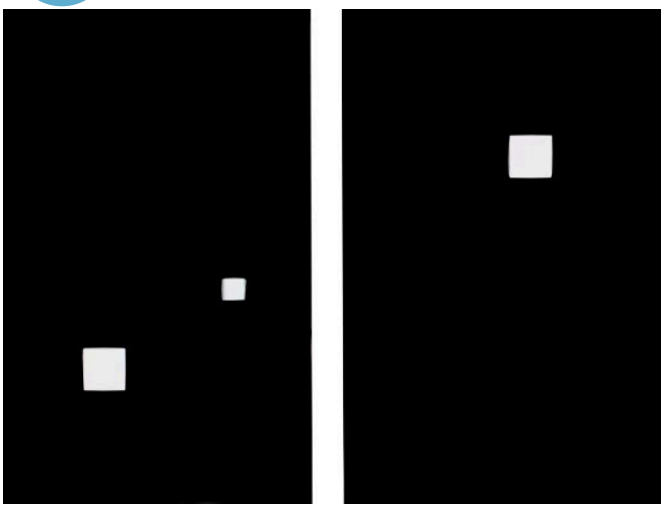


Fig. 5 - Il gioco del Tennis della Odyssey, da cui Bushnell ha preso ispirazione per il suo PONG (foto: Schnurrikowski - fonte: Wikimedia Commons)

Decidono però, prima di metterlo in commercio, di testare il gioco: montano quindi la scheda elettronica all'interno di un cabinato di legno, dove installano anche un comune TV Toshiba comprato in un negozio di elettrodomestici, fissano sul pannello frontale 2 paddle, ed infine adattano la gettoniera di una lavatrice a gettoni per dare il segnale che fa avviare una partita. Bushnell si accorda con Bill Gaddis, uno dei clienti Atari a cui sono forniti i flipper, che acconsente ad installare la macchina di prova nel suo locale, l'Andy Capp's Tavern di Sunnyvale (fig. 4). Il gioco mostra subito il suo potenziale: molti clienti iniziano a frequentare il locale non più solo per consumare e poi eventualmente distrarsi qualche minuto con uno dei flipper ma anche solo per fare una partita a questo nuovo videogioco, chiamato "PONG" dal fatto che simula una partita di ping-pong fra 2 giocatori, presentandosi alle volte anche prima dell'orario di apertura. Dopo alcuni giorni, però, il gioco si blocca: quando Alcorn va a vedere quale possa essere il difetto, trova che la gettoniera è piena di monete e rigetta fuori quelle immesse, non avviando più una nuova partita. Una volta svuotata, il gioco riparte, per la gioia di Bushnell che capisce che quello è un vero affare! Il 29 novembre 1972 PONG viene ufficialmente presentato al pubblico, e subito diviene un successo.

Dopo aver trovato un finanziatore e risolti i problemi di assemblaggio dei primi esemplari, la costruzione dei cabinati prosegue senza sosta, portando un sacco di soldi nelle casse di Atari: basti pensare che ogni singolo arcade PONG viene venduto mediamente a più del doppio del suo costo di produzione, generando quindi un bel profitto.

Il gioco è un successo ed il mercato si allarga: agli inizi del 1973, viene esportato anche all'estero. Alla fine del 1973 si contano 2500 arcade venduti e l'anno successivo questo valore raggiunge quota 8.000: se si considera il periodo, il tipo di prodotto ed il mercato a cui è destinato, questi numeri sono di tutto rispetto.

La causa con Magnavox

Ovviamente tutto questo successo non passa inosservato, né da parte del grande pubblico, agli occhi del quale Atari inizia a costruirsi quella fama di grande produttore di videogiochi che consoliderà di lì a poco, né da qualcuno che si sente derubato della propria idea. Sì, stiamo parlando proprio di Magnavox, che vede troppe similitudini fra il gioco del tennis della sua console Odyssey (fig. 5) e PONG. Agli inizi del 1974 decide perciò di far valere i brevetti registrati da Ralph Baer, lo sviluppatore originale della console (se volete approfondire la storia della Odyssey, vi rimandiamo alla lettura dell'articolo ad essa dedicata sul numero 34 della nostra rivista), e cita in giudizio non solo Atari ma diverse altre società che, nel frattempo, hanno rilasciato dei cloni di PONG. Bushnell ha effettivamente assistito alla presentazione del gioco e le firme di presenza all'evento lo incastrano perciò non vuole rischiare una causa legale che potrebbe prosciugare le casse di Atari e decide di accordarsi extra-corte, pagando la (ragguardevole) cifra di 1,5 milioni di dollari in più rate per poter divenire licenziatario Magnavox. L'accordo, stilato a giugno 1976, prevede anche che per la durata di un anno Atari informi Magnavox di tutti i prodotti messi in commercio: questa clausola pare che sia stata anche

Fig. 6 - Il Tele-Games, la prima versione dell'Home PONG così come viene commercializzata nella catena di negozi Sears (foto: Evan-Amos - fonte: Wikimedia Commons)

Fig. 7 - Il Commodore TV Game 2000K, commercializzato nella sola Europa (foto: José Antonio Ortigueira Amor - fonte: Wikimedia Commons)

uno dei motivi per cui Atari abbia atteso fino al mese di settembre del 1977 per presentare ufficialmente la sua nuova console VCS senza essere costretta a rivelare dettagli della stessa a Magnavox.

La versione domestica

I guadagni vanno discretamente, anche se la concorrenza ha intaccato le vendite: per ovviare a ciò si pensa ad una versione del gioco realizzata su un singolo chip LSI (Large Scale Integration), sia per ridurre i costi di produzione sia anche per rendere più difficoltoso il lavoro di retro-ingegnerizzazione con il quale la concorrenza copia PONG. Lo sviluppo di questo chip richiede tempo (e denaro) e, nel frattempo, Bushnell chiede ai suoi ingegneri di pensare a qualcosa di nuovo, di spremere nuove idee per far fruttare PONG e portare nuovi soldi nelle casse della società. Harold Lee, un altro ingegnere Atari, inizia a maturare l'idea di usare questo nuovo chip LSI per una versione domestica di PONG. Lee lavora con Alcorn allo sviluppo del chip mentre Bob Brown lavora ad un primo

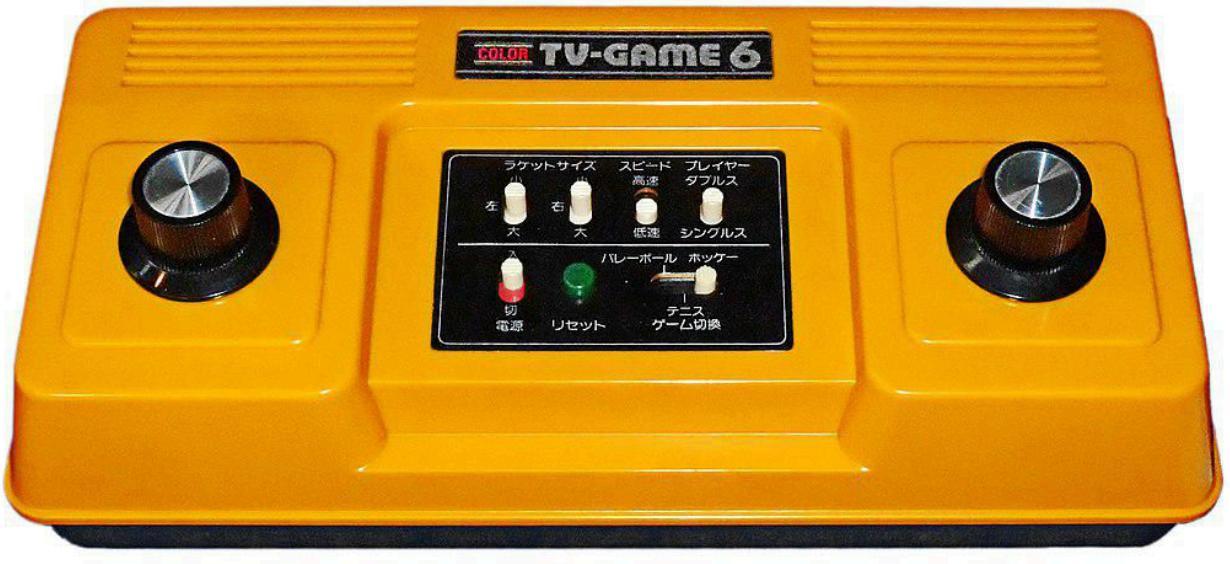


Fig. 8 - l Nintendo Color-TV Game 6 (foto: Chapuisat - fonte: Wikimedia Commons)

prototipo collegabile ad un televisore. Alla fine del 1974 il chip è pronto ed il prototipo anche: Atari contatta quindi Sears per la distribuzione in esclusiva di questa versione domestica del gioco che nel frattempo è stata chiamata "Home PONG". L'Home PONG viene messo in vendita nei negozi della catena Sears come "Tele-Games" (fig. 6) al prezzo di 99 dollari tra ottobre e novembre del 1975, giusto in tempo per le festività natalizie: le circa 150.000 unità fornite a Sears vengono vendute tutte. Nel 1976 l'Home PONG viene messo in commercio anche direttamente da Atari, che ne vende circa 50.000 unità.

Cloni e varianti

Il successo dell'arcade attira la concorrenza: capito che PONG è molto popolare e che questo nuovo mercato dei videogiochi è redditizio, nel giro di poco tempo compaiono sul mercato cloni del gioco che vanno ad intaccare la quota dell'originale Atari. Replicare l'arcade è molto semplice: ciò è dovuto al fatto che il gioco non contiene un computer che esegue un programma bensì è composto da una semplice scheda elettronica fatta di componenti discreti e integrati TTL normalmente reperibili in commercio: basta perciò copiare la scheda e mettere gli stessi componenti ed il gioco è bello e clonato.

Come nel caso dell'arcade, anche la versione domestica va incontro allo stesso problema, nonostante il fatto che sia basata su un singolo chip. I numeri fatti registrare dall'Home PONG spingono diversi produttori di integrati ad offrire i propri chip che contengono copie del gioco.

Fig. 9 - Il Coleco Telstar
(foto: Chris Yarzab - fonte: Wikimedia Commons)

Spesso, i produttori non solo replicano il gioco ma ne offrono anche più varianti, alcune delle quali migliorate. È il caso di MOS Technology, che offre un chip con 4 varianti del gioco, utilizzato dalla Commodore per le sue console 2000K (fig. 7) e 3000H (vendute solo sul mercato europeo). Coleco, che più tardi rilascerà il famoso computer ColecoVision, entra nel mercato dei videogiochi domestici proprio con la console Telstar (fig. 9), venduta in ben 14 modelli via via aggiornati con sempre più versioni migliorate del PONG originale, per un totale di circa 1 milione di unità. Anche Nintendo entra nel mondo dei videogiochi domestici con un clone del PONG denominato Color TV-Game (fig. 8): contando le versioni successive, il Color TV-Game è stato venduto in circa 3 milioni di esemplari, gettando le basi per lo sviluppo del più famoso Nintendo Entertainment System.

La stessa Atari ha rilasciato il suo arcade "Breakout" nel 1976 come versione aggiornata di PONG: a differenza di quest'ultimo, qui un solo giocatore deve rompere un muro posto in alto sullo schermo distruggendo i singoli mattoncini che lo compongono colpendoli con una pallina fatta rimbalzare contro una racchetta che si muove nella parte inferiore del campo di gioco.

L'eredità

PONG è uno di quei giochi iconici: non c'è persona, nel periodo tra la fine degli anni '70 e gli inizi degli anni '80 del XX secolo, che non sia entrato in contatto, per averlo avuto a casa propria o averlo visto da qualche amico, con uno delle centinaia di cloni in circolazione in quel periodo. Ma anche chi non è nato o cresciuto in quel periodo sa riconoscerlo, essendo stato riproposto nel corso degli anni su tante piattaforme, anche in epoca più recente: ad esempio, il gioco è stato portato su diverse console contemporanee e sugli smartphone, a dimostrazione che PONG, a distanza di tutti questi anni dal debutto, resta sempre PONG. Infatti, molte pubblicazioni considerano PONG come il prodotto che ha creato trasformato l'industria dei videogiochi in un affare remunerativo, e l'Home PONG come il videogioco che ha portato al successo commerciale il mercato delle console domestiche. Numerose riviste nel corso degli anni lo hanno collocato nei propri elenchi dei titoli più famosi della storia dei videogiochi, tributando a PONG quel dovuto onore conquistato a suon di monetine e sfide fra amici.

Atari 2600 – Tecnologie AI e Movimenti in gioco

di Giampaolo Moraschi

Un interessante progetto per i nostri Atari 2600 che va ad implementare l'uso delle mani al posto del classico Joystick.

Realizzato da Eran Feit, questo concetto vede ogni singolo movimento a schermo fatto con gli arti superiori. Nessun ausilio del Joystick. Il progetto è in grado di rilevare i gesti del corpo per simulare le direzioni: su, giù, sinistra, destra e le azioni di "fuoco" del pulsante. Servono un po' di dimestichezza con Python e una buona dose di tecnica.

Nella pagina web dello sviluppatore (che potete trovare a questo indirizzo:

https://medium.com/@feitgemel/a-revolutionary-way-to-play-atari-2600-using-ai-faa7d7ee7aa6

sono allegati numerosi esempi e lo schema per il collegamento del progetto.

Il codice è sviluppato in Python:

```
import mediapipe as mp
import cv2
import time

from numpy import hamming
#from activateKeyboard import IKEY, UKEY, OKEY,
COMMAKEY, SPACE
#from activateKeyboard import PressKey, ReleaseKey

#
print("start Joystick")

import RPi.GPIO as GPIO
import time

UP_PIN = 6
LEFT_PIN = 23
RIGHT_PIN = 22
DOWN_PIN = 27
BUTTON_PIN = 12

GPIO.setmode(GPIO.BCM)
GPIO.setup(UP_PIN,GPIO.OUT)
GPIO.setup(LEFT_PIN,GPIO.OUT)
GPIO.setup(RIGHT_PIN,GPIO.OUT)
GPIO.setup(DOWN_PIN,GPIO.OUT)
GPIO.setup(BUTTON_PIN,GPIO.OUT)

# starting point - all LOW
```

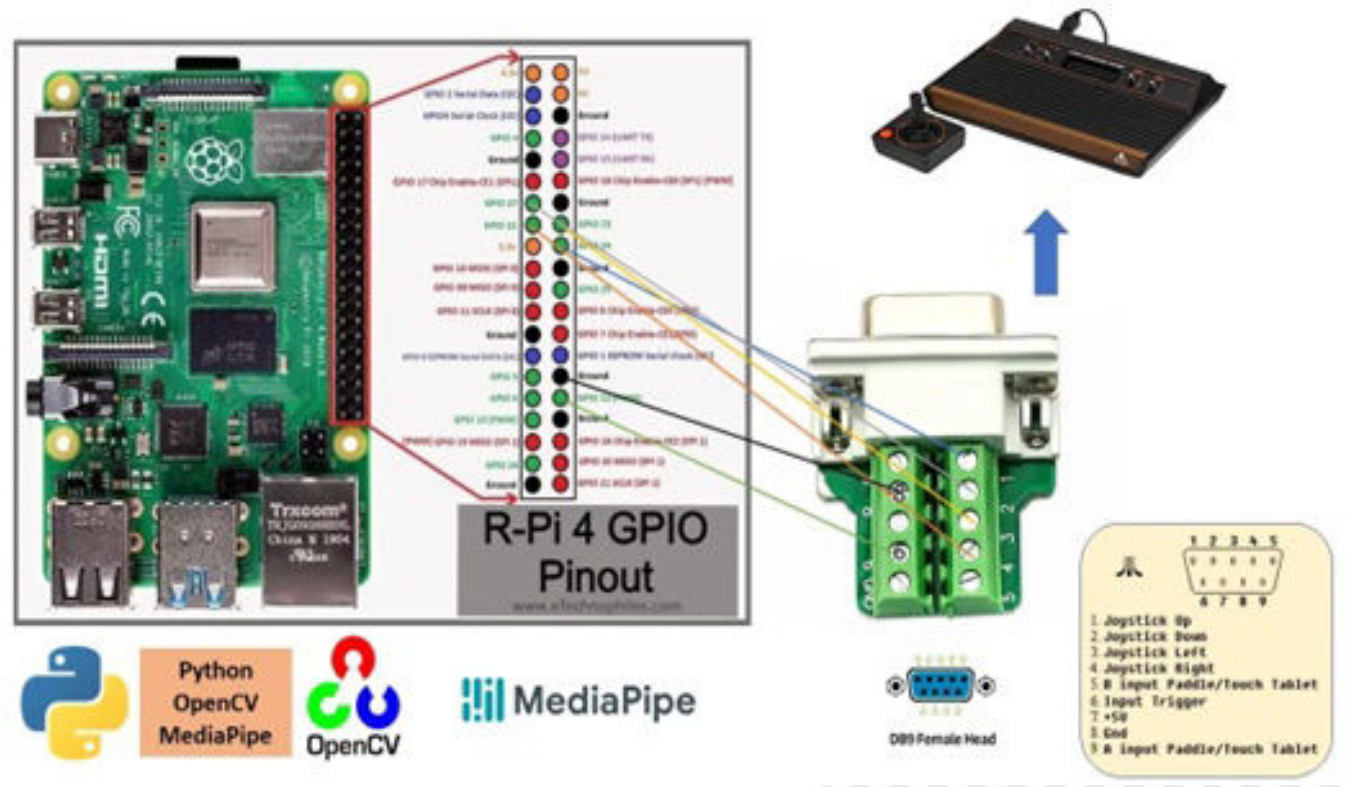


Fig. 1 - Lo schema di collegamento

```
# ========================
GPIO.output(UP_PIN,GPIO.LOW)
GPIO.output(LEFT_PIN,GPIO.LOW)
GPIO.output(RIGHT_PIN,GPIO.LOW)
GPIO.output(DOWN_PIN,GPIO.LOW)
GPIO.output(BUTTON_PIN,GPIO.LOW)

mp_drwaing = mp.solutions.drawing_utils #drawing
uility - Help to render the landmarks
mp_hands = mp.solutions.hands

waitTimePressKey=0.3
difHeight = 100
maxHeight = 150
fireHeight = 80
LastMove=''
fireStatus=False

cap = cv2.VideoCapture(0)
cap.set(3,1280)
cap.set(4,720)

with mp_hands.Hands(min_detection_confidence=0.5,
min_tracking_confidence=0.5,max_num_hands=1)  as
hands:

    while cap.isOpened():

        re, frame = cap.read()

        scale = 75
        w = int(frame.shape[1] * scale / 100)
        h   = int(frame.shape[0] * scale / 100)

        dim = (w,h)

        resized = cv2.resize(frame , dim ,
interpolation = cv2.INTER_AREA)

        # Start the detection :
        # ======================

        # change it to RGB
        image =
cv2.cvtColor(resized,cv2.COLOR_BGR2RGB)

        # flip the image
        image = cv2.flip(image, 1)

        image.flags.writeable = False
        results = hands.process(image)
        image.flags.writeable = True

        centerX = int(w/2)
        centerY = int(h/2)

        gap = 100

        x1 = centerX - gap
        x2 = centerX + gap
        y1 = centerY - gap
        y2 = centerY + gap

          cv2.rectangle(image, (x1, y1), (x2,y2),
(255, 0, 0), 3)

        # center of frame
        #cv2.circle(image, (centerX,centerY) ,
2 , (255,0,0), -1 )

        if results.multi_hand_landmarks:

            for handLMS in
results.multi_hand_landmarks:

#mp_drwaing.draw_landmarks(image,handLMS) # draw
dots - handLMS is the hand index

mp_drwaing.draw_landmarks(image,handLMS,
mp_hands.HAND_CONNECTIONS) # draw connection -
handLMS is the hand index
                for id , lm in
enumerate(handLMS.landmark):
                    h , w , c = image.shape
                     cx , cy = int(lm.x * w) , int
(lm.y * h)
                    #print (id, cx , cy)

                    if id == 4 : # this is the
thumb landmark
                        cv2.circle(image, (cx,cy) ,
3 , (0,255,0), -1 )
                        thumbTipX = cx
                        thumbTipY = cy

                    if id == 8 : # this is the
```

```
indexFingerTip landmark
                    cv2.circle(image, (cx,cy) ,
3 , (0,255,0), -1 )
                    indexFingerTipX = cx
                    indexFingerTipY = cy

                if id == 12 : # this is the
middleFingerTip landmark
                    cv2.circle(image, (cx,cy) ,
3 , (0,255,0), -1 )
                    middleFingerTipX = cx
                    middleFingerTipY = cy

                if id == 16 : # this is the
ringFingerTip landmark
                    cv2.circle(image, (cx,cy) ,
3 , (0,255,0), -1 )
                    ringFingerTipX = cx
                    ringFingerTipY = cy

                if id == 20 : # this is the
pinkeyTips landmark
                    cv2.circle(image, (cx,cy) ,
3 , (0,255,0), -1 )
                    pinkeyTipX = cx
                    pinkeyTipY = cy

                if id == 0 : # this is the wrist
landmark
                    cv2.circle(image, (cx,cy) ,
3 , (0,255,0), -1 )
                    wristX = cx
                    wristY = cy

        fireStatus=False
        GPIO.output(BUTTON_PIN,GPIO.HIGH)

        # check for fire

        #print("abs(wristY-
middleFingerTipY)",abs(wristY-middleFingerTipY))
        #print("abs(wristX-
middleFingerTipX)",abs(wristX-middleFingerTipX))
        #print("abs(pinkeyTipX-
thumbTipX)",abs(pinkeyTipX-thumbTipX))

        if abs(wristY-middleFingerTipY) <
fireHeight and abs(wristX-middleFingerTipX) <
fireHeight and abs(pinkeyTipX-thumbTipX) <
fireHeight:
            fireStatus=True
            GPIO.output(UP_PIN,GPIO.HIGH)
            GPIO.output(LEFT_PIN,GPIO.HIGH)
            GPIO.output(RIGHT_PIN,GPIO.HIGH)
            GPIO.output(DOWN_PIN,GPIO.HIGH)
            GPIO.output(BUTTON_PIN,GPIO.LOW)
            #print('FIRE')
            cv2.rectangle(image, (50, 50),
(250,150), (255, 255, 255), cv2.FILLED)
            cv2.putText(image,'Fire',(70,120),
cv2.FONT_HERSHEY_COMPLEX , 2 , (0,102,0), 5 )

        # Up direction
        if (indexFingerTipY < y1 and
fireStatus==False) :
            #print('UP')
            GPIO.output(UP_PIN,GPIO.LOW)
            GPIO.output(LEFT_PIN,GPIO.HIGH)
            GPIO.output(RIGHT_PIN,GPIO.HIGH)
            GPIO.output(DOWN_PIN,GPIO.HIGH)
            GPIO.output(BUTTON_PIN,GPIO.HIGH)
            LastMove="UP"
            cv2.rectangle(image, (50, 50),
(250,150), (255, 255, 255), cv2.FILLED)
            cv2.putText(image,'UP',(70,120),
cv2.FONT_HERSHEY_COMPLEX , 2 , (0,102,0), 5 )

            #print(str(wristY-middleFingerTipY))

        # Stop direction
        if (indexFingerTipX > x1 and
indexFingerTipX < x2 and indexFingerTipY > y1 and
indexFingerTipY < y2 and fireStatus==False)  :
            #print('STOP')
            GPIO.output(UP_PIN,GPIO.HIGH)
            GPIO.output(LEFT_PIN,GPIO.HIGH)
            GPIO.output(RIGHT_PIN,GPIO.HIGH)
            GPIO.output(DOWN_PIN,GPIO.HIGH)
            GPIO.output(BUTTON_PIN,GPIO.HIGH)
            LastMove="STOP"
            cv2.rectangle(image, (50, 50),
(250,150), (255, 255, 255), cv2.FILLED)
            cv2.putText(image,'STOP',(70,120),
cv2.FONT_HERSHEY_COMPLEX , 2 , (0,102,0), 5 )

            #print(str(wristY-middleFingerTipY))
```

```
            # down direction
            if (indexFingerTipY > y2  and
fireStatus==False)  :
                #print('DOWN')
                GPIO.output(UP_PIN,GPIO.HIGH)
                GPIO.output(LEFT_PIN,GPIO.HIGH)
                GPIO.output(RIGHT_PIN,GPIO.HIGH)
                GPIO.output(DOWN_PIN,GPIO.LOW)
                 GPIO.output(BUTTON_PIN,GPIO.HIGH)
                LastMove="DOWN"
                cv2.rectangle(image, (50, 50),
(270,150), (255, 255, 255), cv2.FILLED)
                 cv2.putText(image,'DOWN',(70,120),
cv2.FONT_HERSHEY_COMPLEX , 2 , (0,102,0), 5 )
              #print(str(middleFingerTipY-wristY ))

            # left direction
            if (indexFingerTipX < x1  and
fireStatus==False)  :
                #print('LEFT')
                GPIO.output(UP_PIN,GPIO.HIGH)
                GPIO.output(LEFT_PIN,GPIO.LOW)
                GPIO.output(RIGHT_PIN,GPIO.HIGH)
                GPIO.output(DOWN_PIN,GPIO.HIGH)
                 GPIO.output(BUTTON_PIN,GPIO.HIGH)
                LastMove="LEFT"
                cv2.rectangle(image, (50, 50),
(270,150), (255, 255, 255), cv2.FILLED)
                 cv2.putText(image,'LEFT',(70,120),
cv2.FONT_HERSHEY_COMPLEX , 2 , (0,102,0), 5 )

            # right direction
            if (indexFingerTipX > x2 and
fireStatus==False)  :
                #print('RIGHT')
                GPIO.output(UP_PIN,GPIO.HIGH)
                GPIO.output(LEFT_PIN,GPIO.HIGH)
                GPIO.output(RIGHT_PIN,GPIO.LOW)
                GPIO.output(DOWN_PIN,GPIO.HIGH)
                 GPIO.output(BUTTON_PIN,GPIO.HIGH)
                LastMove="RIGHT"
                cv2.rectangle(image, (50, 50),
(270,150), (255, 255, 255), cv2.FILLED)
                cv2.putText(image,'RIGHT',(70,120),
cv2.FONT_HERSHEY_COMPLEX , 2 , (0,102,0), 5 )

        image = cv2.cvtColor(image,cv2.COLOR_RGB2BGR)
         cv2.imshow('image',image)
         cv2.moveWindow('image',40,0)

         if cv2.waitKey(10) & 0xFF == ord('q'):
             break

cap.release()
cv2.destroyAllWindows()
```

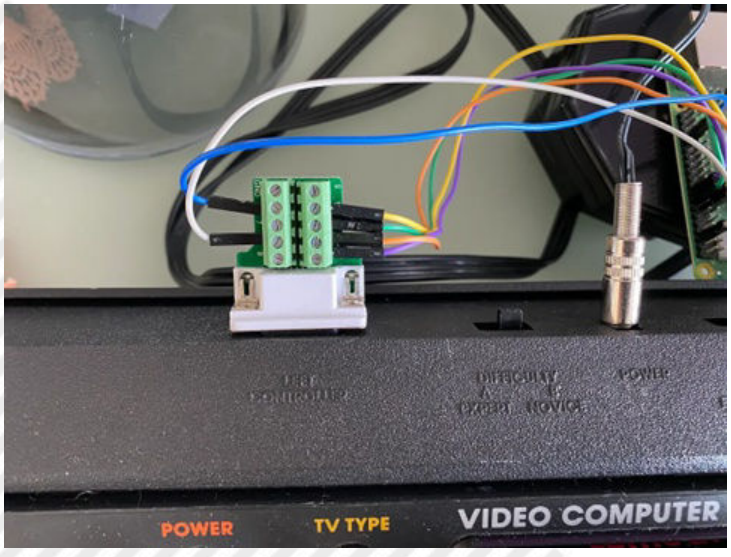


Fig. 2 - Il progetto terminato

Per poter essere sviluppato necessita inoltre di diversi componenti e l'autore nella pagina ha inserito quello che serve per provare a realizzarlo.

Cercheremo di seguire meglio nei prossimi numeri e sulla pagina facebook lo sviluppo futuro e magari di testarlo quanto prima per una recensione.

THE SPECTRUM

Come fare breccia nei cuori degli innamorati dei prodotti di Sir Clive

di Giampaolo Moraschi

Premetto subito. Personalmente lo comprerò. Molti puristi inorridiranno per questa scelta, ma nel 2024 direi che sarebbe anche ora di smetterla di fare a gara a cosa sia real retrò o meno retrò. Lo trovo un prodotto utile e versatile. Detto questo una pre impressione di The Spectrum puramente personale.

Plaion e Retro Games LTD, lo hanno annunciato ufficialmente a fine agosto 2024 suscitando clamore su tutta la rete e non solo.

Una ricreazione moderna del bellissimo ZX Spectrum del 1982. Il modello è stato poi "toccato con mano" alla Gamescon 2024 e sappiamo che il lancio in Europa avverrà il 22 novembre e che è disponibile il pre-ordine per 99 eurini e pochi spiccioli.

Lo ZX Spectrum per il sottoscritto ha sempre avuto un fascino incredibile per estetica e per alcuni titoli ludici all'interno. E' da considerarsi uno dei simboli della rivoluzione tech degli anni 80 e sono particolarmente contento di questa sua "rinascita".

Mi ha molto colpito il lancio di una versione completa. Nessuna mini versione ne futura maxi versione. Qui abbiamo un prodotto con 48 titoli pre installati, quattro porte USB, uscite HDMI a 720p e una replica perfetta della famosa tastiera "in gomma" proprio come il prodotto del 1982.

Mossa azzeccatissima per chi vuole già "smanettare" con il Basic di questa macchina e non solo utilizzarla come console casalinga.

Nella confezione finale non sarà presente un joystick, ma sarà possibile attaccare qualsiasi tipo di controller usb o pendrive alle porte ed utilizzarlo senza problemi.

I titoli presentati sono i seguenti:

- Alien Girl (Skirmish Edition)
- Ant Attack
- Army Moves
- Auf Wiedersehen Monty
- Avalon
- Bobby Bearing
- Cosmic Payback
- Devwill Too ZX
- Exolon
- Fairlight
- Firelord
- Football Manager 2
- Freddy Hardest
- The Great Escape
- Head Over Heels
- Highway Encounter

Fig. 1 - THE SPECTRUM nella sua futura confezione e manuale

Fig. 2 - THE SPECTRUM: la tastiera è riprodotta con tanto di tasti “in gomma”

- The Hobbit
- Horace Goes Skiing
- Jack the Nipper
- Knot in 3D
- The Lords of Midnight
- Manic Miner
- Match Day II
- Movie
- Nodes of Yesod
- Penetrator
- Phantis (Game Over II)
- Pheenix
- Pyracurse
- Quazatron
- Robin of the Wood
- Saboteur! Remastered
- Shovel Adventure
- Skool Daze
- Snake Escape
- Spellbound
- Starquake
- Starstrike II

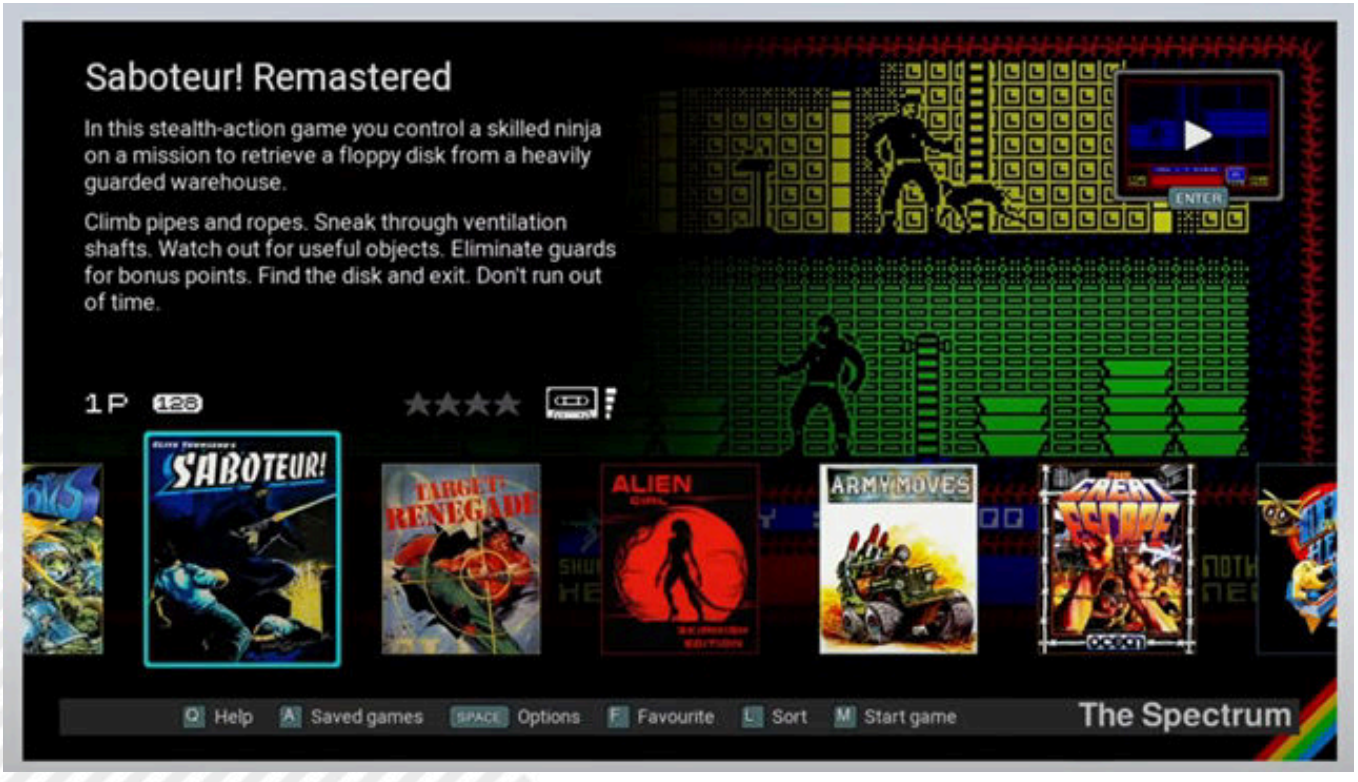


Fig. 3 - Il carosello d’introduzioni con i48 titoli pre installati

Fig. 4 - Sarà possibile inserire i nuovi titoli tramite USB

- El Stompo
- Stonkers
- Target: Renegade
- TCQ
- Technician Ted – The Megamix
- Tenebra
- Trashman
- The Way of the Exploding Fist
- Wheelie
- Where Time Stood Still

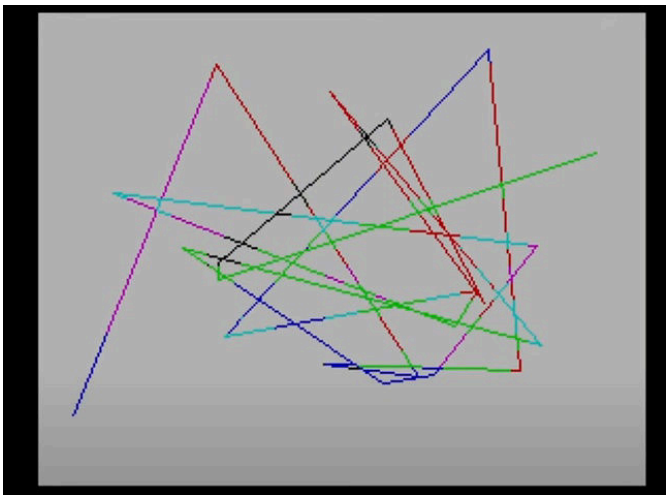


Fig. 5 - Non mancherà nemmeno il BASIC

Una bella lista con qualche chicca che The Hobbit che rigiocherò volentieri. Tutti i titoli saranno presentati con licenza ufficiale.
Saranno emulati tutti i modelli di ZX Spectrum dai 16k ai 128k, si potrà mappare i controlli dei giochi con interfaccia apposita e ovviamente sarà possibile caricare ulteriori titoli vecchi e nuovi tramite pendrive.
Così sarà possibile giocare ad alcuni piccoli capolavori come Seraphima, Alien:Neoplasma o Mighty Final Fight.

Non vedo l'ora personalmente di metterci le mani sopra e testarlo per una recensione più approfondita.
Ben vengano queste produzioni come le riedizioni del 2600 o il futuro 7800+, il mercato è in espansione e lo spazio c'è per tutti.

Grazie Retro Games LTD per le foto e il press kit.

Il Nintendo 64

di Bimbomillo

Il Nintendo 64 (abbreviato in N64) è una console per videogiochi a 64 bit prodotta da Nintendo dal 1996 al 2002 e succeduta al Super Nintendo Entertainment System (SNES).

In Italia la console è stata messa in vendita nel 1997 dal Gruppo Giochi Preziosi sotto il marchio GIG.

Quando ormai l'era del Super NES era avviata definitivamente alla fine, le grandi software house iniziarono a pensare alla successiva generazione di console, e mentre molte case optarono per il passaggio ai 32 bit, la Nintendo scelse invece di passare direttamente ai 64 bit e così, dopo alcuni anni di sviluppo, nacque l'Ultra 64, poi cambiato in Nintendo 64, poiché la Konami fece uso di questo termine per la divisione Ultra Games.

La scelta di Nintendo di usare anche per questa console il sistema a cartucce, giustificata dal desiderio di scongiurare la pirateria informatica, fu oggetto di alcune critiche da parte di tutti coloro che attendevano un prodotto avveniristico al pari della rivale PlayStation; la PlayStation infatti poteva riprodurre anche CD musicali oltre ai giochi, offrendo quindi un intrattenimento multiforme che colpì particolarmente i consumatori.

Inoltre il basso costo di produzione dei CD-ROM consentì alla Sony di rimettere in vendita i giochi non più di punta a prezzi ridotti, cosa che invece Nintendo non poteva fare a causa dell'alto prezzo di fabbricazione delle cartucce.

Il Nintendo 64 segnò di fatto una vera e propria rivoluzione nel campo dei videogiochi, proponendo la prima vera esperienza tridimensionale di gioco con titoli di elevata qualità e tuttora autentiche pietre miliari del loro genere, ed ebbe riscontri positivi in tutto il mondo superando la quota di 30 milioni di unità vendute.

Tuttavia, negli ultimi anni di vendita fu sorpassato dalla rivale PlayStation, che stabilì un record vendendo oltre 70 milioni di unità, ma con risultati nettamente superiori a quelli del Sega Saturn le cui vendite raggiunsero poco più di 9 milioni di unità.

Lo sviluppo di applicazioni del Nintendo 64 terminò il 30 aprile 2002.

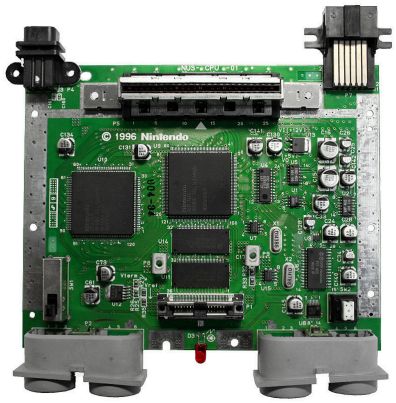
La macchina si presenta come una scatola rettangolare di colore nero (ma durante gli anni uscirono anche edizioni speciali caratterizzate da colori diversi) con gli spigoli arrotondati e che presenta sul lato superiore una fessura per l'inserimento delle cartucce contenenti i giochi e sulla parte frontale 4 connettori per i gamepad, senza comprare periferiche esterne.

La console funziona mediante cartucce che vengono direttamente inserite nella macchina. Questo sistema consente di immagazzinare meno dati rispetto ai CD-ROM, ma permette tempi di caricamento minimi, possibilità di salvataggio senza ausilio di memory card, presentando un sistema di riconoscimento delle cartucce tramite un circuito elettronico chiamato Checking Integrated Circuit affiancato da un bus di tipo Serial Peripheral Interface.

Presenta quattro uscite per i gamepad, in modo tale che si possa giocare con titoli multiplayer per quattro giocatori senza dover comprare supporti aggiuntivi come il Multitap di Sony

Sotto il gamepad è disponibile uno slot che consente di inserire periferiche addizionali. Davanti alla console è presente uno slot per inserire un nuovo banco di memoria RAM. A differenza del suo predecessore (il SNES), il Nintendo 64 non può fornire in uscita un segnale video RGB, e, nella versione europea, nemmeno S-Video. Questo significa che un N64 europeo ha come miglior segnale video possibile in uscita il video composito. Sotto alla console è presente uno slot utilizzato per collegare la console con il Nintendo 64DD, espansione hardware uscita solo in Giappone.

Durante il ciclo di vita del Nintendo 64 vennero prodotti diversi componenti aggiuntivi ufficiali per la console:
Controller Pak: una memoria da 256 IO che si inseriva sotto il pad, divisa in 123 "pagine". Successivamente Nintendo, accortasi della minima dimensione di questa scheda di memoria decise di crearne altre versioni dalla dimensione da I fino a 4 MB.
Expansion Pak: si tratta di un banco di 4 MB di RDRAM (Rambus DRAM) che va a affiancarsi al banco originale da 4 MB montato on-board, questo consente di raggiungere gli 8 MB di RAM, permettendo di utilizzare alte definizioni di gioco e molti altri miglioramenti. Sebbene sia supportato da diversi giochi, alcuni di questi possono funzionare solo se l'Expansion Pak è stato installato sulla console, mentre altri necessitano dell'Expansion Pak per poter attivare alcune funzionalità extra del gioco.
Rumble Pak: è un accessorio che inserito nel gamepad vibra durante la partita in seguito a eventi generati nell'ambiente di gioco. Questo accessorio ormai è considerato uno standard disponibile per tutte le console delle ultime generazioni. Si tratta di fatto di un motorino alimentato da 2 pile ministilo.
Transfer Pak: è un accessorio che inserito nel controller permette di trasferire i dati di gioco tra il Nintendo 64 e il Game Boy, la console portatile Nintendo. Realmente, questo accessorio aveva una sola utilità, ossia quella di poter trasferire sul Nintendo 64 le immagini catturate con la Game Boy Camera ma in seguito fu adoperato nei giochi di Pokémon Stadium per passare dati dal Game Boy al Nintendo 64.



La scheda madre del Nintendo 64 con la CPU, l'RCP (il Reality Co-Processor) e la RDRAM

Nintendo 64 Game Paks aperto e chiuso

Specifiche tecniche

La scheda madre del Nintendo 64 con 1a CPU, l'RCP (il Reality Co-Processor) e 1a RDRAM Central Processing Unit:
• NEC VR4300-64 bit da 4,6 milioni di transistor @93,75 MHz
• 24 kB cache di primo livello
• Architettura MIPS / R4300i RISC
• Capacità matematica totale della CPU: 93,0 milioni di operazioni al secondo
• Processo di fabbricazione a 0,35 um (micrometri)
Graphics Processing Unit:
• Coprocessore Silicon Graphics-RPC Reality 64 bit @62,5 MHz Unità vettoriale per interi a 8-bit
• Rendering grafico di picco pari a 150.000 poligoni al secondo con tutte le elaborazioni, 600.000 poligoni monocromi al secondo.

Elaborazione:
• Z-buffering
• Anti-aliasing
• Texture mapping
• Filtro lineare (bilineare e trilineare)
• Mip-mapping
• Gouraud shading
• Fillrate in pixel da 30 megasamples al secondo con Z-buffer
• 16.7 milioni di colori (32.768 sullo schermo)
• Risoluzioni: 320 x 240/640x480 pixels
• Performance GPU: 600 MegaFlops

Memoria:
• 4 MByte RDRAM
• System Bandwidth @562,5 MB/secondo
• Latenza di 640 nanosecondi

Sonoro:
• 16-bit stereo
• 100 canali PCM lineare (massimo 16-24 di elevata qualità). Ogni canale occupa per intero un ciclo della CPU
• Frequenza di campionamento a 48,0 kHz
• Formati supportati: MIDI, MP3, ADPCM e Tracker

Supporti di Memorizzazione:
• Cartucce elettroniche da 4 a 64 MByte

Controller con stick analogico, funzione di vibrazione e ingresso per memory card (Controller Pak)

Input / Output:
• 4 porte per controller
• Porta d'espansione per Expansion Pak

Riferimenti:

- *https://en.wikipedia.org/wiki/Nintendo_64*
- *https://it.wikipedia.org/wiki/Nintendo_64*

Tutti i marchi rappresentati appartengono ai legittimi proprietari.

Donkey Kong su PC128 Olivetti Prodest
Il nuovo retro-progetto di Paolo Cattaneo

di Paolo Cattaneo e Carlo Nithaiah Del Mar Pirazzini

Paolo Cattaneo è uno sviluppatore talentuoso. Ammaliato dalle potenziali di ugBASIC, ha tirato fuori dal cilindro un port davvero interessante che mostra come "si può fare..." sia la vera formula dietro alla nuova ondata del retrogaming. Ma lasciamo la parola a lui e addentriamoci nel suo mondo.

Il PC 128 Olivetti Prodest

Possedere un PC128 Olivetti Prodest, nel 1986 o giù di lì, equivaleva a far parte di una riserva indiana.

Tramite qualche oscuro e bizzarro accordo commerciale, la Olivetti aveva messo le mani sul francese Thomson MO6 (che in patria fu un fiasco pazzesco) e lo aveva rivenduto spacciandolo come produzione propria dopo avergli appioppato un bel logo nostrano ad hoc e aver convinto il direttore di Commodore Italia a passare dalla loro parte.

Per qualche altrettanta oscura ragione, il tentativo riuscì a persuadere molti più genitori di quanti non furono invece oltralpe, e la macchina finì così sulla scrivania di parecchi adolescenti della penisola, tra cui appunto il sottoscritto, che ingenuamente vollero credere di poter vivere le stesse gioie degli amici possessori di Commodore 64 o similari.

Cosa spinse la Olivetti a commercializzare quella macchina così superata nelle specifiche persino per l'anno in cui usciva, non si è mai capito, ma tra una vasta gamma di micro computer concorrenti che si fregiavano di infinite librerie di giochi - molti dei quali conversioni di titoli arcade all'ultimo grido - il PC 128 offriva un parco software decisamente più limitato sia in termini qualitativi che quantitativi.

I pochi nomi altisonanti presenti sulla macchina per noi lettori di Zzap! erano quelli di Arkanoid o Green Beret, e vi basteranno due secondi su youtube per constatarne la resa: gli sviluppatori facevano il possibile nel breve tempo che avevano, ma la macchina purtroppo era quella che era.

Parliamo di un Motorola 6809E a 1Mhz con nessun tipo di chip ausiliario, nemmeno per l'audio, gestito da un povero beeper a 1bit in sync col processore; quest'ultimo, come se non bastasse, presenta una fatale mancanza di operazioni da registro a registro: è necessario passare attraverso la RAM per confrontare i registri A e B (entrambi a 8 bit), il che fornisce un confronto di almeno 10 cicli di CPU, quando l'ottimale sarebbe farlo in 1 o 2 cicli (come tutte le operazioni immediate tipo CMPA, SUBB, ecc.). Anche lo scambio dei registri A e B costa ben 8 cicli.

Per farla breve, muovere le cose a schermo su questa macchina significa fare i conti con una lentezza disarmante se non si limita adeguatamente il numero di pixel che vanno ridisegnati ad ogni frame.

E quindi, tornando alla riserva indiana, i provetti avventurieri dell'home gaming come me si resero presto conto che la loro macchina non avrebbe mai potuto equiparare nemmeno uno Spectrum in termini di velocità e azione: lo Spectrum aveva senz'altro una resa grafica obsoleta, ma faceva girare Chase H.Q. e Metro Cross con una fluidità che un PC128 non si sarebbe mai sognata.

Chiunque lo ha posseduto avrà senz'altro provato almeno una volta la frustrazione di cui sto parlando, quel non poter invitare a casa vostra l'amichetto col C64 perché vi avrebbe semplicemente riso in faccia.

Fig. 1 - Olivetti Prodest PC 128 S in tutto il suo splendore

Molti acquirenti del PC 128, una volta accortisi della faccenda, decisero semplicemente di cambiare macchina. Nel mio caso non si poteva.

Mio padre aveva appositamente scelto il PC 128 affinché imparassi a programmare. Lui non avrebbe mai sopportato l'idea di comprarmi una "console travestita da computer" che "mi avrebbe solo tolto ore allo studio", e visto che l'azienda per la quale lavorava aveva iniziato a impartire dei corsi di BASIC interni, fu proprio lui a passarmi i primi rudimenti del linguaggio. A conti fatti, devo ringraziarlo per la sua lungimiranza: non so se scegliendo un altro computer non sarei diventato il programmatore a tempo pieno che sono ancora oggi, ma di certo quella riserva indiana mi fece presto scoprire la bellezza di quest'arte.

In tutto ciò, fece anche la sua buona parte il residente SIMIV BASIC 128, realizzato da Microsoft e molto versatile, praticamente un antesignano di quello che sarà poi il GWBasic. Insomma: accettato il fatto che non era la macchina giusta per giocare, un programmatore aveva parecchio di che divertirsi, anche in virtù di più di 100KB a disposizione per il suo listato!

Naturalmente, i miei anni da adolescente sul PC 128 non arrivarono mai a farmi produrre niente di davvero eclatante, anche perché la programmazione di videogiochi di un certo tipo comportava la conoscenza di argomenti che raramente erano trattati in modo così tecnico nel nostro paese. Ma, anche quando un paio di anni dopo smisi di usarlo per passare a un IBM 8088, decisi che non mi sarei mai separato da questo mio primo computer: prima o poi ci avrei fatto un gioco "come si deve", a voler rivendicare quella parte di me che aveva sperato in una macchina comparabile con le altre in grado di dare grandi soddisfazioni ludiche.

L'era moderna e l'avvento di ugBasic

La storia riprende circa una decina d'anni fa, quando iniziano a sorgere le prime comunità online dedicate al retro sviluppo su Thomson e quindi anche sul gemello francese del PC 128.

Finalmente è l'occasione per avere accesso ad un mare di documentazione dell'epoca che per tutto quel tempo era sempre stata confinata in Francia, tra manuali di assembler e schemi tecnici con la descrizione di ogni banco di memoria. In un certo senso, è stato come ricominciare da zero, ma in modo più professionale: mi

Fig. 2 - Asmshooter – una demo per uno shooter scritto interamente in ASM 6809

dedico all'ASM 6809, e con un pò di fatica arrivo a realizzare qualche piccola demo che finalmente gira fluida.

Mi rendo anche conto che lo sviluppo di un intero gioco in ASM sarebbe veramente molto impegnativo in termini di tempo e metodo: un piccolo errore scoperto troppo tardi rischia di dover riconsiderare tutto il lavoro, e gli errori in ASM sono anche piuttosto difficili da scovare man mano che il programma si fa sempre più grosso. Inoltre, la scena dei partecipanti e degli utenti dei vari homebrew su questa macchina non è particolarmente folta, quindi in questo periodo continuo solo a sviluppare piccole demo, alcune per le challenge dei gruppi facebook di retro programmazione.

Finchè, nel 2021, arriva quella che per me è stata la svolta decisiva: **ugBasic**, il framework realizzato da **Marco Spedaletti**, finalmente è in grado di compilare per il PC 128. E' una svolta perché permette al programmatore di concentrarsi unicamente sulla business logic senza che si faccia scoraggiare dalla lunghezza e dall'osticità dell'assembler: non è più necessario, ad esempio, scrivere le sequenze di disegno che mettono il valore nell'accumulatore e poi nella cella della memoria video desiderata. Il listato viene scritto in forma di alto livello, il caricamento e il disegno degli asset vengono semplificati enormemente dal framework, ma le prestazioni sono quelle del linguaggio macchina.

A quel punto, senza neanche più le barriere date dal metodo di programmazione, c'erano davvero i mezzi per

```
UPDATESHOTS
          * Check for start
          LDA MSTSHT
          CMPA #01
          BNE UPDSHTS
          JSR STARTSHOT
UPDSHTS
          * Update *
          LDX #$00
UPDATESHOT
          LDA SHOTS,X
          CMPA #$02
          LBLO UPDNXTSHT   .active
          LDD SHOTS+6,X    .get pos
          STD SHOTS+8,X    .upd pos
          LDA SHOTS+1,X    .frame
          CMPA #$00        .advance frame
          BEQ ADVSHTPOS
          LDA #$00
          STA SHOTS+1,X
          LDU #PLRSHT2     .load sprite
          BRA UPDSHTDONE
ADVSHTPOS
          * Advance position
          LDA #$01
          STA SHOTS+1,X    .advance frame
          LDU #PLRSHT1     .load sprite
          LDD SHOTS+6,X
          ADDD #$0001
          STD SHOTS+6,X
          LDA SHOTS+10,X   .offset x
          * Check right screen border
          CMPA #$26
          BGE SHTBORDER
          INC SHOTS+10,X   .increment
          BRA UPDSHTDONE
SHTBORDER
          LDA #$01
          STA SHOTS,X
          BRA UPDNXTSHT
UPDSHTDONE
```

Fig. 3 - ASMCODE: Una porzione di codice per scrivere la demo di asmshooter

realizzare qualcosa di sostanzioso. Ho così ripensato al desiderio di realizzare una conversione convincente di un gioco famoso su questa macchina, e ho individuato quelli che sarebbero dovuti essere i criteri di selezione del candidato. Per motivi di performance, il gioco doveva essere un monoschermo e con un massimo di 800 pixel ridisegnabili ad ogni frame; per motivi di experience, il gioco doveva avere più livelli; la colonna sonora di fondo durante il gameplay doveva essere assente o comunque non ricoprire un ruolo importante; infine, il gioco non doveva essere già stato convertito in precedenza.

La scelta avvenne quando una sera, incuriosito dalla meccanica con cui Mario percorreva le pendenze in Donkey Kong, provai a replicare la stessa cosa mentre smanettavo con ugBasic, e ci riuscii prima di quanto pensassi (il che, conoscendomi, è decisamente strano XD). Nel giro di un paio d'ore avevo tutto il primo livello pronto, e appariva piuttosto convincente, quindi decisi che evidentemente non ci sarebbe stato candidato più idoneo, a patto di presentarlo solo come demo tecnica per non scomodare eventuali implicazioni legali (ma su questo ci torno dopo).

Donkey Kong

Finora ho descritto il PC 128 come una macchina lenta, ma va sottolineato che la sua palette di colori resta a mio avviso la più bella mai apparsa su un computer 8 bit, con una tavolozza di ben 4096 sfumature da scegliere per comporre la gamma dei 16 che andranno a video.

Questa eccellenza tecnica ha permesso di restituire una grafica piuttosto fedele a quella del coin op, a patto ovviamente di scendere a compromessi con le specifiche tecniche originali. Tutti gli sprites del gioco da bar sono stati più o meno ridimensionati, ad esempio Kong nell'arcade arriva fino a 43px in lunghezza mentre qui sta sempre in una cornice di 40x32, Mario sta in una cornice di 16x16 ma di fatto è alto 1 pixel in meno rispetto all'originale, e specialmente tutti i livelli hanno dovuto subire un compattamento. Mario è appunto più piccolo del dovuto per dare più respiro all'occhio nel primo livello 1, dove tutti

```
' tapis roulant livello 3
IF (curLvl == 02 AND curPage == 01) THEN: reposPlayer[]:
    IF (yOnMap == 161) THEN: mvPlrH[-01]: ENDIF
    IF (yOnMap == 090) THEN: IF (plrX < 80) THEN: mvPlrH[01]: ELSE: mvPlrH[-01]: ENDIF: ENDIF
    IF (yOnMap == 062) THEN: mvPlrH[-01]: ENDIF
ENDIF

plrFrm = plrSFrm
IF (KEY PRESSED(#LEFT))  THEN: ftL = 13: ftR = 05: mvPlrH[-01]: plrSFrm = 04: plrJFrm = 07: ENDIF
IF (KEY PRESSED(#RIGHT)) THEN: ftL = 05: ftR = 13: mvPlrH[01]:  plrSFrm = 00: plrJFrm = 03: ENDIF
IF (timeHammer == 00) THEN
IF (KEY PRESSED(#SPACE)) THEN: isJumping = 01: jumpCount = -04: plrFrm = plrJFrm:
    isJX = 0: IF (KEY PRESSED(#LEFT) OR KEY PRESSED(#RIGHT)) THEN: isJX = 1: ENDIF
    jmpnt = 01: GOSUB progressJump: ENDIF: ENDIF
```

Fig. 4 - Un po' di ugCODE, uno snipper del codice di ugBasic, serve a far veder la differenza di scrittura rispetto ad ASM (alto livello vs Basso livello)

i 6 piani di pendenza sono stati avvicinati allo scopo di non sacrificarne nessuno. L'arcade ha risoluzione 224x256x256 colori che ho dovuto tradurre nella modalità 160x200x16 colori del PC 128. Questa modalità presenta tre vantaggi: 1) non c'è nessun color clash, 2) utilizza due pagine video interscambiabili che evitano il flickering, e 3) tutte le coordinate a schermo possono stare ognuna in un byte, il che rende più veloce i calcoli. Le due pagine video presentano una peculiarità fastidiosa: va tenuta traccia non di una ma di ben due posizioni indietro per ogni aggiornamento, e questo inevitabilmente porta a usare più variabili di quelle che servirebbero per cancellare e ridisegnare.

La memoria della macchina viene utilizzata dividendosi in un massimo di 26Kb per il codice e quasi tre banchi da 16Kb per la grafica (di cui uno solo per la splash screen). Ogni livello del gioco presenta caratteristiche particolari: il primo deve far sì che Mario percorra le pendenze, e lo stesso devono fare i barili; il secondo ha una piattaforma mobile che deve trasportare Mario; il terzo ha dei tapis roulant che devono rallentare o accelerare Mario; il quarto ha piattaforme che scompaiono quando Mario ci passa sopra. Per gestire i livelli ho utilizzato una matrice di valori per ognuno di esso: lo schermo di gioco è 160x200 quindi una divisione logica in 10x20 tiles da 16, 10 pixel sarebbe stato un buon compromesso rispetto a una 20x20 con tiles larghe solo 8pixel, sicuramente più dettagliata ma ben più costosa in memoria.

Questo tiles rappresenta il modo in cui ho diviso i livelli in caselle per questa conversione.

Ogni valore della matrice specifica quindi la coordinata Y dove mario deve poggiare i piedi (quindi Mario verrà disegnato a quella coordinata, meno 16 pixel più in alto). Ecco che allora la matrice del primo livello presenta valori sfalsati di una unità, supponiamo che mario sia con i piedi appoggiati alla coordinata Y 192, la riga in quel punto della matrice sarà una cosa tipo 192, 191, 190, 189, 188 ecc ... e noi lo vedremo salire andando da sinistra verso destra. Analogamente, i livelli in piano presentano righe con valori tutti uguali (192, 192, 192, ecc.).

Le scale sono invece gestite con array: andavano bene anche le matrici ma sono più lente come accesso, nell'ottica di dover fare tutto ottimizzato ho puntato su un massimo di 6 scale quindi due array da 6 valori che indicano X e Y

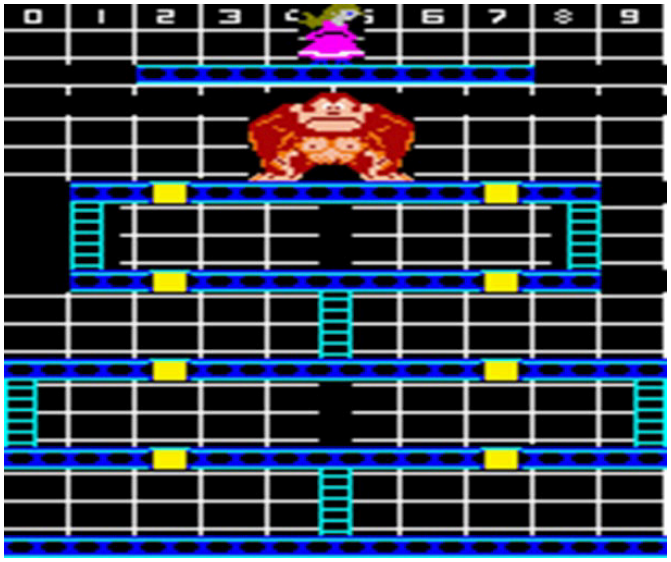


Fig. 5 - Questo tiles rappresenta il modo in cui ho diviso i livelli in caselle per questa conversione

di inizio e fine scala. I barili devono controllare la presenza di scale sopra o sotto di loro ogni qual volta che si spostano, quindi i cicli di controllo devono essere ridotti al minimo funzionale. Il detect di Mario costa meno perché viene fatto solo su evento pressione tasto direzionale su o giu, e una volta passato in stato "sulle scale" ci rimane finchè spostandosi non detecta la coordinata di fine di una scala, tornando così allo stato precedente.

Per la piattaforma mobile del secondo livello, ho fatto un po' il contrario: è lei a controllare che Mario sia sopra di essa, e quando questo avviene è lei a ordinare di spingerlo in alto come posizione una volta fatti i dovuti controlli di stato. La molla che rimbalza non è 16x16 ma 8x16 pixel così da ottimizzare la velocità di gioco.

I tapis roulant invece funzionano invocando la routine di camminata direzionale senza che il giocatore abbia effettivamente premuto alcun tasto per muovere Mario: questo viene fatto ovviamente solo quando Mario viene rilevato con i piedi appoggiati sopra i rulli, ma specialmente questo viene fatto solo in una delle due pagine video, così da usare l'altra per lasciare al giocatore un frame neutro dove poter far spostare Mario su propria volontà, cosa che altrimenti in quel caso lo vedrebbe restare fermo sul posto (tutti i movimenti orizzontali di Mario sono infatti di 4 pixel).

Le piattaforme a scomparsa sono gestite con uno status che si attiva quando Mario ci cammina sopra. Una volta che lo status è attivato, se la coordinata di Mario non è più inclusa nei limiti di quella piattaforma, essa scompare dal livello (disegno il blocco nero e viene modificata la

Fig. 6 - La schermata del titolo ci ricorda di "comprare i giochi NINTENDO"... Non facciamoli arrabbiare!

relativa matrice).

La splash screen è stata realizzata adattando il flyer originale giapponese del 1981, portato alla risoluzione di 160x200x16 e rilavorato nei pixel e nei colori.

Per quanto riguarda l'audio, non c'è molto da dire: al momento ugBasic non supporta comandi per il sonoro, ma supporta codice ASM inline, per cui ho utilizzato quella soluzione. Ho dapprima "suonato" i vari motivetti direttamente in BASIC 128 su emulatore, ho segnato i valori e li ho riportati nel codice ASM per settare e suonare il beeper. I suoni salto e jingle del bonus sono eseguiti uno step per frame con una partitura contenuta in un array.

Ho cercato di includere tutti e 4 i livelli dell'arcade come priorità principale, a discapito di alcune mancanze o inaccuratezze che con Marco stiamo cercando di sistemare. Vedremo di dare ai nemici di fuoco un comportamento più fedele a quello dell'arcade, e di inserire i bonus e le scenette che separano i vari livelli. Se poi tutto va come deve, speriamo anche di includere la possibilità di giocare con gli sprite di Luigi e Popeye.

Distribuzione e guidelines legali Nintendo

Ho dato un'occhiata alla (lunga) lista dei fan game (tipo il mio) che Nintendo ha cercato di bloccare, spesso riuscendoci, e mi sono fatto l'idea che, a differenza di un po' tutte le altre game companies, Nintendo prenda queste cose molto, molto sul serio.

E a costo di apparire noioso vi dirò: posso benissimo capirli. Non è tanto una questione di lucro, è che i loro personaggi portano comunque il nome Nintendo: pertanto, se rappresentati in modo non adeguato, Nintendo viene rappresentata in modo non adeguato., ed è Nintendo l'unica che deve decidere se e quando il modo è adeguato. Penso che la vedano così, e allora rispetto questa loro visione. Se poi considero l'odissea che il colosso di Kyoto ha dovuto subire proprio riguardo a questo gioco, partito per rimediare al fallimento di Radar Scope e finito per portarli in tribunale contro l'Universal rischiando di farli chiudere, capisco bene perché proteggano i loro prodotti con il rigore di un samurai.

Con la scomparsa di Mr. Iwata (sigh!) Nintendo non ha mai rinnovato le guidelines per gli homebrew, pertanto sono costretto a non distribuire i binari nè i sorgenti per poter giocare questa demo. Spero che presto qualcosa da Kyoto si muova e che un giorno anche voi possiate testare con mano. Nel frattempo, ovviamente, vi invito a giocare a Donkey Kong nei mille altri modi (legali) possibili.

E ringrazio voi di Retro Magazine World per la bellissima opportunità di questo racconto! :)

Aggiornamento

Dopo averlo promesso, alla fine l'ha fatto! Paolo è riuscito ad inserire in game la possibilità di scegliere Luigi e Popeye.

Il titolo è davvero bellissimo!

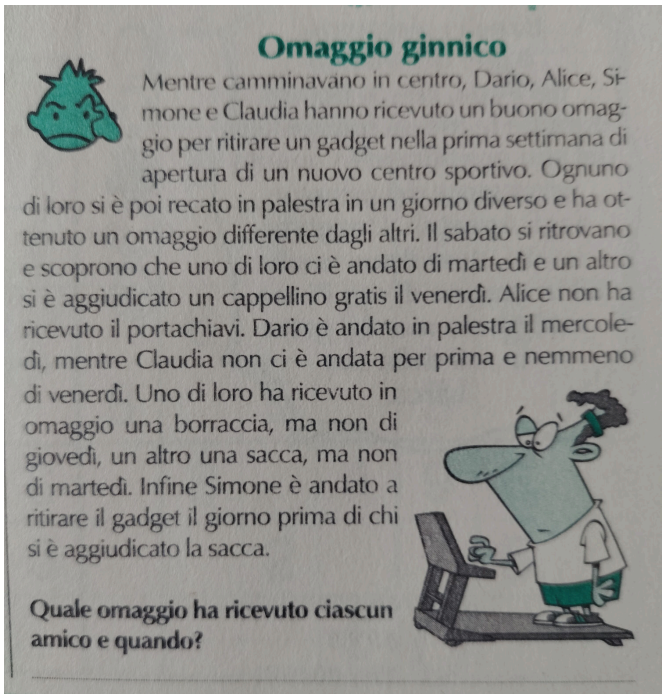
Il C64 incontra l'Intelligenza Artificiale

di Eugenio Rapella

Dario, Alice, Simone e Claudia hanno ricevuto un buono omaggio per ritirare un gadget nella prima settimana di apertura di un nuovo centro sportivo. Ognuno di loro si è poi recato in palestra in un giorno diverso e ha ottenuto un omaggio differente dagli altri. Il sabato si ritrovano e scoprono che uno di loro ci è andato di martedì e un altro si è aggiudicato un cappellino gratis il venerdì. Alice non ha ricevuto il portachiavi. Dario è andato in palestra il mercoledì, mentre Claudia non ci è andata per prima e nemmeno di venerdì, Uno di loro ha ricevuto in omaggio una borraccia, ma non di giovedì; un altro una sacca, ma non di martedì. Infine Simone è andato a ritirare il gadget il giorno prima di chi si è aggiudicato la sacca.

Quale omaggio ha ricevuto ciascun amico e quando?

Sulle riviste italiane, quesiti come questo (in fig.1 il testo originale che proviene dal n. 987 di "Settimana SUDOKU") vengono spesso etichettati come "quesiti di logica". Se devo essere sincero, li trovo piuttosto noiosi e certo non perdo tempo a cercare di risolverli.



Omaggio ginnico

Mentre camminavano in centro, Dario, Alice, Simone e Claudia hanno ricevuto un buono omaggio per ritirare un gadget nella prima settimana di apertura di un nuovo centro sportivo. Ognuno di loro si è poi recato in palestra in un giorno diverso e ha ottenuto un omaggio differente dagli altri. Il sabato si ritrovano e scoprono che uno di loro ci è andato di martedì e un altro si è aggiudicato un cappellino gratis il venerdì. Alice non ha ricevuto il portachiavi. Dario è andato in palestra il mercoledì, mentre Claudia non ci è andata per prima e nemmeno di venerdì. Uno di loro ha ricevuto in omaggio una borraccia, ma non di giovedì, un altro una sacca, ma non di martedì. Infine Simone è andato a ritirare il gadget il giorno prima di chi si è aggiudicato la sacca.

Quale omaggio ha ricevuto ciascun amico e quando?

Si può fare qualcosa con il nostro Commodore? Il Basic del C64 non sembra il linguaggio più adatto per affrontare il nostro quesito (che ne dite del Prolog?).

Mi sono divertito a trovare un modo per descrivere la situazione al C64; fatto questo, l'ho lasciato libero di lavorare e … non mi ha deluso!

```
100 dim p1(24),p2(24):print chr$(5)
110 for k=1 to 24:read p1(k):p2(k)=p1(k):next
120 data 1234,1243,1324,1342,1423,1432
130 data 2134,2143,2314,2341,2413,2431
140 data 3124,3142,3214,3241,3412,3421
150 data 4123,4132,4213,4231,4312,4321
160 dim n$(4),day$(4),gad$(4),day(4),gad(4)
170
n$(1)="dario":n$(2)="alice":n$(3)="simone":n$(4)="cla
udia"
180
day$(1)="mart.":day$(2)="merc.":day$(3)="giov.":day$(
4)="ven."
190 gad$(1)="cappellino":gad$(2)="portachiavi"
200 gad$(3)="borraccia":gad$(4)="sacca"
210 for h=1 to 24:for k=1 to 24
220 v1=int(p1(h)/10):day(1)=p1(h)-10*v1
230 v2=int(v1/10):day(2)=v1-10*v2
240 day(4)=int(v2/10):day(3)=v2-10*day(4)
250 v1=int(p2(k)/10):gad(1)=p2(k)-10*v1
260 v2=int(v1/10):gad(2)=v1-10*v2
270 gad(4)=int(v2/10):gad(3)=v2-10*gad(4)
280 for a=1 to 4
290 if gad(a)=1 then x=a
300 next
310 if day(x)<>4 then 510
320 if gad(2)=2 then 510
330 if day(1)<>2 then 510
340 if day(4)=1 or day(4)=4 then 510
350 for a=1 to 4
360 if gad(a)=3 then x=a
370 next
380 if day(x)=3 then 510
390 for a=1 to 4
400 if gad(a)=4 then x=a
```

```
410 next
420 if day(x)=1 then 510
430 for a=1 to 4
440 if gad(a)=4 then x=a
450 next
460 if day(x)-1<>day(3) then 510
470 print
480 for w=1 to 4
490 print n$(w),day$(day(w)),gad$(gad(w))
500 next
510 next:next
```

Tanto per cominciare, carichiamo i nomi dei quattro amici nel vettore N$(..); i nomi dei giorni della settimana (solo quelli che servono) in DAY$(..) e, infine, i nomi dei gadget nel vettore GAD$(..).
I vettori sono dimensionati alla 160 e le assegnazioni effettuate nelle istruzioni dalla 170 alla 200.
Dunque:
DARIO > 1; ALICE > 2; SIMONE > 3; CLAUDIA > 4.

MARTEDÌ > 1; MERCOLEDÌ > 2; GIOVEDÌ > 3; VENERDÌ > 4.

CAPPELLINO > 1; PORTACHIAVI > 2;
BORRACCIA > 3; SACCA > 4.

Ogni personaggio va in palestra in un giorno differente e riceve un omaggio differente. Prima di considerare altre limitazioni, questo significa che una "possibile soluzione" sarà qualcosa del tipo:

```
NOME:      1     2     3     4
++++++++++++++++++++++++++
GIORNO:    3     1     2     4
GADGET:    4     1     3     2
```

In questo esempio, DARIO (1) va in palestra il GIOVEDÌ (3) e riceve la SACCA (4); ALICE (2) ci va al MARTEDÌ (1) e riceve il CAPPELLINO (1) ecc.
La prima riga, quella dei nomi, è fissata: 1,2,3,4.
Nella seconda riga ci sarà una permutazione dei numeri da 1 a 4, così ognuno si reca in palestra in un giorno differente. Anche nella terza riga ci sarà una permutazione delle cifre da 1 a 4 in modo che gli amici ricevano omaggi differenti.

L'idea di base è quella di abbinare ciascuna delle permutazioni dei giorni (le permutazioni di quattro elementi sono 24) con ciascuna delle permutazioni dei gadget.
Dovremo poi trovare il modo di descrivere le limitazioni segnalate nel testo in modo da "filtrare" le "candidate soluzioni" (che, all'inizio, sono ben 24*24=576).
Alla fine verranno stampate solo quelle che rispettano i requisiti e, se il problema è ben posto, ... ne rimarrà esattamente una: la soluzione.
Le 24 permutazioni dei numeri 1,2,3,4 sono caricate nei vettori P1(..) e P2(..) all'inizio del programma (istruzioni 100-150). Avremmo potuto chiedere al C64 di generarle lui stesso, ma sono talmente poche che è più comodo e veloce fornirle direttamente.
Per comodità, non ho utilizzato P1(0) e P2(0); alla fine del FOR-NEXT dell'istruzione 110, avremo P1(1)=1234, P1(2)=1243, P1(3)=1324, ..., P1(24)=4321, e lo stesso per P2(..).

Il programma vero e proprio ha inizio all'istruzione 210 con i due FOR nidificati che si chiudono con i due NEXT dell'ultima istruzione (la 510).
Per ogni H da 1 a 24, si preleva la permutazione di posto H dal vettore P1(..) e le quattro cifre che la compongono vengono assegnate al vettore DAY(..) che è stato dimensionato alla 160. Questa assegnazione avviene nelle tre istruzioni 220-240; se, ad esempio, P1(H)=2341, avremo V1=INT(P1(H)/10)=INT(234,1)=234.
DAY(1)=P1(H)-10*V1=2341-10*234=1.
V2=INT(V1/10)=INT(234/10)=23; DAY(2)=234-10*23=4;
DAY(4)=INT(V2/10)=INT(23/10)=2
e DAY(3)=V2-10*DAY(4)=23-10*2=3.
In definitiva, se P1(H)=2341, si ottiene DAY(1)=1, DAY(2)=4, DAY(3)=3, DAY(4)=2: le cifre della permutazione vengono travasate nel vettore DAY(..).
Lo scopo del vettore DAY(..) è quello di associare il giorno della settimana al personaggio: se DAY(2)=3 significa che il giorno associato ad ALICE (2) è il GIOVEDÌ (3).
Discorso del tutto analogo per la permutazione estratta dal vettore P2(..): se P2(K)=1243, avremo GAD(1)=3, GAD(2)=4, GAD(3)=2 e GAD(4)=1. Col vettore GAD(..) associamo al personaggio un oggetto: se GAD(4)=1 significa che a CLAUDIA (4) associamo il CAPPELLINO (1).

Se e quando vengono raggiunte, le istruzioni 470-500

stampano i nomi degli amici (N$(..)) associati ai nomi dei giorni (DAY$(..)) corrispondenti al DAY(..) e ai nomi degli oggetti (GAD$(..)) corrispondenti a GAD(..).
Se eliminate tutte le istruzioni dalla 280 alla 460 e lanciate il programma, il C64 elenca diligentemente i nomi degli amici abbinati alle permutazioni dei giorni e, a ciascuna di queste, vengono abbinate tutte le possibili permutazioni dei gadget. Si tratta di ben 24*24=576 elenchi: uno di questi elenchi è la soluzione del nostro problema.
Come trovarlo? Qui viene il bello: le segnalazioni contenute nel testo si trasformano in una serie di IF: non appena una richiesta non viene rispettata, si salta immediatamente alla 510 per l'esame di una nuova candidata. Alla 470 si giungerà solo se tutte le richieste sono soddisfatte!

Seguiamo il testo:
[1] QUALCUNO SI È AGGIUDICATO IL CAPPELLINO (1) DI VENERDÌ (4):

```
280 for a=1 to 4
290 if gad(a)=1 then x=a
300 next
310 if day(x)<>4 then 510
```

Il ciclo 280-300 consente di determinare il numero X della persona che ha il cappellino. Alla 310 si controlla: se il giorno associato a X non è il venerdì la nostra condizione non è soddisfatta e si può andare direttamente alla 510: la "candidata" non può essere la soluzione.

++

[2] ALICE (2) NON HA RICEVUTO IL PORTACHIAVI (2):

```
320 if gad(2)=2 then 510
```

Ovvero: se il gadget associato ad Alice è proprio il portachiavi, quella in esame non può essere la soluzione; passiamo ad un'altra candidata.

++

[3] DARIO (1) È ANDATO IN PALESTRA IL MERCOLEDÌ (2):

```
330 if day(1)<>2 then 510
```

Se il giorno associato a Dario non è mercoledì ... non ci siamo e si va oltre.

++

[4] CLAUDIA (4) NON VA IN PALESTRA IL PRIMO GIORNO (1) E NEPPURE DI VENERDÌ (4):

```
340 if day(4)=1 or day(4)=4 then 510
```

Se ci è andata di martedì o venerdì, questa non può essere la soluzione.

++

[5] CHI HA RICEVUTO LA BORRACCIA (3), NON L'HA RICEVUTA DI GIOVEDÌ (3):

```
350 for a=1 to 4
360 if gad(a)=3 then x=a
370 next
380 if day(x)=3 then 510
```

Al termine del ciclo 350-370, X contiene il numero della persona destinataria della borraccia. Se il giorno corrispondente a X è proprio giovedì, la richiesta non è rispettata e dobbiamo andare oltre.

++

[6] CHI HA RICEVUTO LA SACCA (4), NON L'HA RICEVUTA DI MARTEDÌ (1):

Operando come nel passaggio precedente:

```
390 for a=1 to 4
400 if gad(a)=4 then x=a
410 next
420 if day(x)=1 then 510
```

++

[7] SIMONE (3) È ANDATO A RITIRARE IL GADGET IL GIORNO PRIMA DI CHI SI È AGGIUDICATO LA SACCA (4):

```
430 for a=1 to 4
440 if gad(a)=4 then x=a
450 next
460 if day(x)-1<>day(3) then 510
```

Al termine del ciclo 430-450, X contiene il numero della persona che ha avuto la sacca. DAY(X)-1 è il giorno precedente a quello utilizzato da X; se questo giorno non è quello corrispondente a Simone, niente da fare, si va oltre…

++++++++++++++++++++++++++++++++++++++

Dato il RUN, il C64 impiega circa un minuto per fornire una soluzione e un ulteriore minuto per completare l'analisi senza aggiungere altre soluzioni. La soluzione proposta dal Commodore è unica e coincide con quella segnalata nella rivista (vedi fig. 2).

Ho approfittato del programma per un piccolo esperimento: cosa succede se una (o più) condizioni vengono ignorate? A rigor di logica, se si elimina una condizione, si dovrebbe ottenere oltre la soluzione originale, almeno una soluzione alternativa.
Se eliminiamo la condizione [1], il C64 produce complessivamente tre soluzioni; se invece eliminiamo la [3] le soluzioni diventano sei. Cancellando la [5], il C64 segnala due soluzioni; eliminando la [7] le soluzioni sono cinque.
La cosa curiosa è che, se si elimina la [2], il C64 fornisce solo la soluzione originale: questo significa che l'informazione "Alice non ha ricevuto il portachiavi" non è una informazione essenziale, è … ridondante.
Lo stesso avviene se si elimina la condizione [4] oppure la [6], ma anche se si eliminano contemporaneamente [4] e [6]! Una sola soluzione (la solita) anche se si ignorano contemporaneamente la [2] e la [6], mentre se si cancellano sia la [2] che la [4], oppure [2]+[4]+[6], si ottengono due soluzioni (i più masochisti potranno provare altre combinazioni … io me ne guardo bene).

Ho sottoposto a ChatGPT il testo originale del quesito, ho ottenuto una risposta molto articolata che si concludeva con

Soluzione finale:
• Martedì: Simone – Borraccia.
• Mercoledì: Dario – Portachiavi.
• Giovedì: Claudia – Sacca.
• Venerdì: Alice – Cappellino.
Questa soluzione rispetta tutte le condizioni del problema

Ho segnalato a ChatGPT che la sua soluzione non rispettava l'ultima condizione ("Simone è andato a ritirare il gadget il giorno prima di chi si è aggiudicato la sacca"); l'Intelligenza Artificiale mi ha risposto:

Mi scuso per l'errore. Rivediamo attentamente il problema, assicurandoci di rispettare tutte le condizioni e di trovare una soluzione corretta.

La cosa è andata avanti per un bel po': ChatGPT proponeva una soluzione errata; io segnalavo l'errore e ChatGPT procedeva ad una nuova proposta con delle variazioni, ma senza progressi.
Alla fine ho segnalato a ChatGPT la soluzione trovata dal Commodore e l'ho invitata a verificare che questa sì rispettava tutte le richieste.
Ecco la risposta finale di ChatGPT:
"Grazie per la tua pazienza e per avermi guidato verso la soluzione corretta".
Che il nostro amato Commodore desse una mano all'I.A. forse non ve lo aspettavate…

Addendum

Questa parte non dice granché di nuovo; la inserisco per completezza e si può senz'altro saltarla.
I giorni della settimana citati nel testo sono quelli da

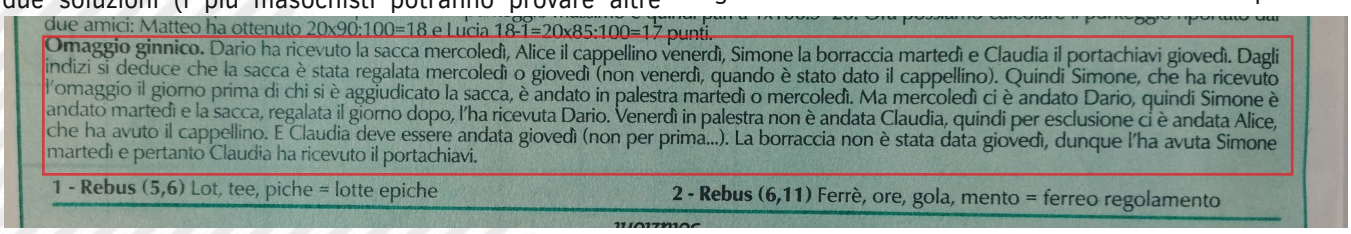

due amici: Matteo ha ottenuto 20x90:100=18 e Lucia 18-1=20x85:100=17 punti.
Omaggio ginnico. Dario ha ricevuto la sacca mercoledì, Alice il cappellino venerdì, Simone la borraccia martedì e Claudia il portachiavi giovedì. Dagli indizi si deduce che la sacca è stata regalata mercoledì o giovedì (non venerdì, quando è stato dato il cappellino). Quindi Simone, che ha ricevuto l'omaggio il giorno prima di chi si è aggiudicato la sacca, è andato in palestra martedì o mercoledì. Ma mercoledì ci è andato Dario, quindi Simone è andato martedì e la sacca, regalata il giorno dopo, l'ha ricevuta Dario. Venerdì in palestra non è andata Claudia, quindi per esclusione ci è andata Alice, che ha avuto il cappellino. E Claudia deve essere andata giovedì (non per prima...). La borraccia non è stata data giovedì, dunque l'ha avuta Simone martedì e pertanto Claudia ha ricevuto il portachiavi.
1 - Rebus (5,6) Lot, tee, piche = lotte epiche
2 - Rebus (6,11) Ferrè, ore, gola, mento = ferreo regolamento

Fig. 1 - La soluzione cosí come appariva sulla rivista

martedì a venerdì (il sabato è escluso visto che è il giorno in cui i ragazzi si ritrovano tutti insieme). Si tratta di quattro giorni, così come quattro sono gli amici e quattro i gadget assegnati: probabilmente questa è l'interpretazione corretta ed è quella che ho utilizzato nella trattazione precedente. A rigor di logica, si potrebbe inserire anche "lunedì": il fatto che non venga mai citato non significa che qualcuno non possa essere andato in palestra di lunedì e sia uno degli altri giorni della settimana a rimanere scoperto.
Per scrupolo, ho provato a variare il programma in questo senso modificando la numerazione relativa ai giorni: lunedì>1; martedì>2; mercoledì>3; giovedì>4: venerdì>5.

Ecco il risultato:

```
100 dim p1(24),p$(120):print chr$(5):kx=1
110 for k=1 to 24:read p1(k):next
120 for k=1 to 24: m$=str$(p1(k))
130 for ww=1 to 5
140 n$=left$(m$,ww)+"5"+right$(m$,5-ww)
150 p$(kx)=n$:kx=kx+1:next ww
160 next k
170 data 1234,1243,1324,1342,1423,1432
180 data 2134,2143,2314,2341,2413,2431
190 data 3124,3142,3214,3241,3412,3421
200 data 4123,4132,4213,4231,4312,4321
210 dim n$(4),day$(5),gad$(4),day(5),gad(4)
220
n$(1)="dario":n$(2)="alice":n$(3)="simone":n$(4)="clau
dia"
230    day$(1)="lun.":day$(2)="mart.":day$(3)="merc."
240 day$(4)="giov.":day$(5)="ven."
250 gad$(1)="cappellino":gad$(2)="portachiavi"
260 gad$(3)="borraccia":gad$(4)="sacca"
270 for h=1 to 120:for k=1 to 24
280 for tt=1 to 5
290 day(tt)=val(mid$(p$(h),tt+1,1)): next tt
300 v1=int(p1(k)/10):gad(1)=p1(k)-10*v1
310 v2=int(v1/10):gad(2)=v1-10*v2
320 gad(4)=int(v2/10):gad(3)=v2-10*gad(4)
330 for a=1 to 4
340 if gad(a)=1 then x=a
350 next
360 if day(x)<>5 then 610
370 if gad(2)=2 then 610
380 if day(1)<>3 then 610
390 if day(4)=1 or day(4)=5 then 610
400 xa=0
410 for a = 1 to 4
420 if day(a)=1 then xa=1
430 next
440 if xa=0 and day(4)=2 then 610
450 for a=1 to 4
460 if gad(a)=3 then x=a
470 next
480 if day(x)=4 then 610
490 for a=1 to 4
500 if gad(a)=4 then x=a
510 next
520 if day(x)=2 then 610
530 for a=1 to 4
540 if gad(a)=4 then x=a
550 next
560 if day(x)-1<>day(3) then 610
570 print
580 for w=1 to 4
590 print n$(w),day$(day(w)),gad$(gad(w))
600 next
610 next:next
```

Le permutazioni dei cinque giorni diventano 5!=120 e le ho ottenute inserendo la cifra 5 in ciascuna delle 24 permutazioni dei numeri 1-2-3-4, nei cinque modi possibili (istr. 120-160). Ad esempio, la prima permutazione 1234 origina i primi cinque elementi di P$(..): 51234, 15234, 12534, 12354, 12345 (la funzione STR$(NUMERO) trasforma il NUMERO in una stringa anteponendo uno "spazio").
I vettori DAY$(..) e DAY(..) vengono dimensionati al valore 5 (non vengono utilizzati DAY$(0) e DAY(0)).
Alla 270 si aprono i due cicli FOR del programma principale e il ciclo FOR relativo alle permutazioni dei giorni va da 1 a 120. La permutazione viene caricata in DAY(..) estraendo i singoli caratteri (con MID$(..)) poi trasformati in cifre (con VAL(..)).
IL resto del programma non si discosta molto da quello già presentato; naturalmente le istruzioni con DAY(..)=N sono state modificate con la nuova numerazione dei giorni. L'unica variazione sostanziale sta nella condizione "Claudia non è andata in palestra per prima". Nel primo programma questa condizione coincideva con "Claudia non è andata di martedì". Ora, oltre a "Claudia è andata di lunedì", va

esclusa un'altra situazione: "se nessuno è andato in palestra il lunedì, Claudia ci va di martedì". Questo si riflette nelle istruzioni:

400 xa=0
410 for a = 1 to 4
420 if day(a)=1 then xa=1
430 next
440 if xa=0 and day(4)=2 then 610

Nella situazione che si sta esaminando, la variabile XA, inizialmente nulla, viene portata a "uno" se qualcuno è andato in palestra il lunedì. Se questo non avviene e Claudia va di martedì (istr. 440), allora Claudia sarebbe la prima ad andare in palestra e la situazione analizzata non potrebbe essere la soluzione (si va al NEXT dell'istr. 610).

Nel primo programma, il gruppo di istruzioni all'interno dei due FOR-NEXT annidati veniva eseguito 24*24=576 volte; ora viene eseguito 24*120=2880 volte per cui il tempo di elaborazione è, naturalmente, maggiore; circa dieci minuti per trovare la prima soluzione e altri sei per completare l'elaborazione. Anche in questa versione del problema, il C64 trova la soluzione e dimostra che è unica. Non merita una standing ovation?

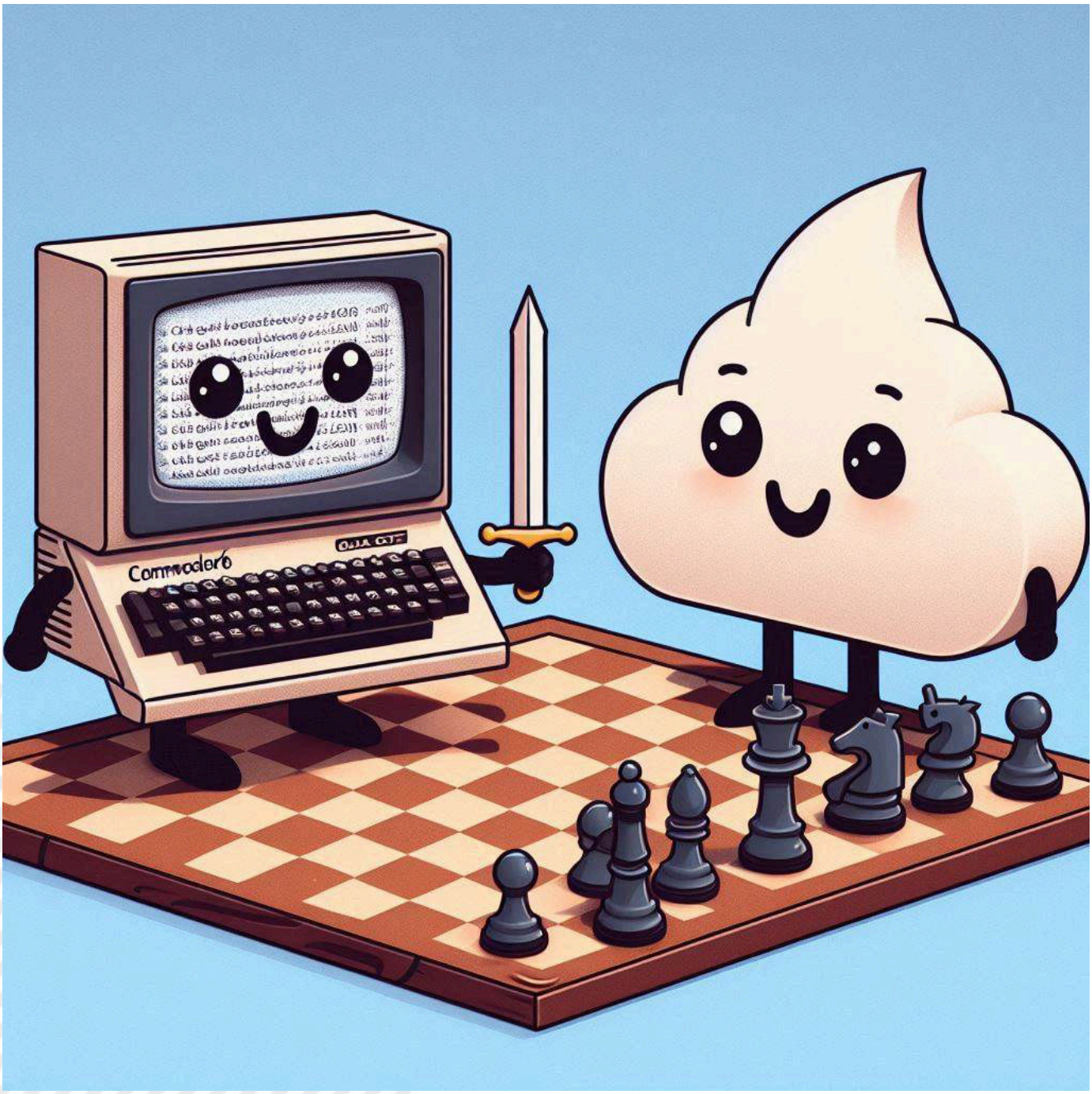


Fig. 2 - L'immagine della sfida immaginata dall'IA di Bing

Zena

di Javier Gonzalez - Traduzione: Fiorentini Francesco

Il VIC1001 è una versione giapponese del VIC20, uscito qualche mese prima, nel 1980. Si tratta sostanzialmente di un VIC20 NTSC.

Il gioco Zena per il VIC1001 era un type-in pubblicato in una rivista giapponese chiamata I/O, numero di ottobre 1983. La rivista può essere trovata (per il momento) su archive.org; contiene belle foto di strani micro giapponesi e altro.

Invece del solito listato BASIC, questo è un listato binario. L'articolo fa riferimento alla cartuccia di monitoraggio del codice macchina VIC1213. Ci sono anche alcune istruzioni su come eseguire il codice. Fortunatamente è facile trovare il d64 online.

L'articolo contiene alcune spiegazioni sul funzionamento interno del gioco. Ho provato a copiare il testo dal pdf e incollarlo in Google Translator con risultati mediocri.

Il programma si avvia anche su un VIC20 PAL, il gioco è giocabile. Funziona su una macchina con memoria espansa a +3K. A quanto pare è ispirato a un gioco chiamato Xevious. Si può trovare nel forum VIC20 su Internet (forum Denial).

Tastiera

Ho avuto problemi a giocare con l'emulatore, perché il gioco usa i tasti due punti (:) e punto e virgola (;) per muovere la nave e il tasto shift per sparare. Questo non funziona con la mia tastiera, perché i caratteri "due punti" e "punto e virgola" sono sullo stesso tasto fisico.

Ho deciso di modificare il programma per cambiare i tasti.

Per prima cosa ho verificato l'uso dei registri della matrice della tastiera. Ho usato il comando hunt del monitor VICE per cercare di trovare la sequenza LDA $9111 negli indirizzi della memoria RAM. Questo ha dato 0 risultati. Poi ho cercato solo l'indirizzo e il comando ha dato un risultato non correlato.

In BASIC mi è sempre piaciuto il trucco WAIT653,1 per aspettare che l'utente premesse il tasto SHIFT. Il fatto che il gioco usi il tasto SHIFT mi ha fatto sospettare che l'autore avesse usato le routine del KERNAL per la tastiera. Ho cercato quindi 653, ci sono stati 1 o 2 risultati, uno dei quali era esatto.

Una volta individuata la routine di input da tastiera, è facile cambiare la pressione dei tasti due punti e punto e virgola in O e P. Come bonus, ho scoperto che era facile applicare una patch per il fuoco automatico. È sufficiente sovrascrivere il test del rilascio del tasto SHIFT prima dello sparo successivo. L'autofire si ottiene premendo SHIFT/LOCK.

```
...
.C:0e09 AD 8D 02 LDA $028D <- 653, indirizzo SHIFT
.C:0e0c F0 0E BEQ $0E1C
.C:0e0e A5 05 LDA $05 <- L'indirizzo di memoria $05 viene utilizzato per testare il rilascio del tasto SHIFT
.C:0e10 D0 0C BNE $0E1E <- per il fuoco automatico cambiare questa istruzione in BIT
...
.C:0e1c 85 05 STA $05
...
.C:0e1f A5 C5 LDA $C5 <- codice dell'ultimo tasto premuto
.C:0e21 C9 2D CMP #$2D <- codice di pressione dei due punti
.C:0e23 F0 0B BEQ $0E30
.C:0e25 C9 16 CMP #$16 <- codice di pressione del punto e virgola
.C:0e27 D0 0C BNE $0E35
...
```

È possibile trovare i codici dei tasti premuti eseguendo

questo piccolo programma BASIC:

```
10 PRINT PEEK(197) : GOTO10
```

L'emulazione non è come l'hardware reale

Stavo giocando come un vero asso. Raggiunsi senza sforzo il boss finale, ma l'emulazione si bloccò. All'inizio ho sospettato che i tre byte che avevo cambiato avessero incasinato qualcosa. Dopo un po' di volte, ho aperto il monitor VICE e ho provato a seguire un loop infinito che si è rivelato insignificante.

```
.C:12e3 20 45 13 JSR $1345
.C:12e6 9D A7 03 STA $03A7,X
```

La copertina di PIO numero 11 del 1985

```
C:12e9 D0 05 BNE $12F0
.C:12eb BD A2 03 LDA $03A2,X
.C:12ee F0 F3 BEQ $12E3
```

All'indirizzo 1345$ ho trovato qualcosa a cui dare la colpa. Si trattava di un generatore di numeri casuali. Sembra che l'autore abbia utilizzato un registro speculare relativo ai paddles.

```
.C:199f A5 FB LDA $FB
.C:19a1 6D 18 90 ADC $9018
.C:19a4 85 FB STA $FB
.C:19a6 60 RTS
```

Grazie ai miei precedenti tentativi di tradurre l'articolo, sono stato in grado di capire che si trattava di un generatore di numeri casuali e non di un metodo di controllo alternativo per usare i paddles.

$1994 numero casuale

Sull'emulatore PEEK($9018) produce sempre 0. Ho cambiato l'indirizzo di $9007, la linea raster. Ora il gioco ha un numero abbastanza elevato di nemici sullo schermo.

Ho deciso di controllare $9018 sulla mia macchina VIC20 NTSC. Sull'hardware reale questo indirizzo funziona effettivamente come generatore di numeri casuali. Non ne conosco il motivo reale.

Il bug

Sembra esserci un bug nel codice originale. Occasionalmente viene visualizzato un simbolo nella colonna più a sinistra. Potrebbe essere legato alla gestione del cursore da parte del kernel. Ma non sembra interferire con il gameplay e potrebbe essere difficile da rintracciare, quindi mi chiamo fuori.

Numero di vite

Nella spiegazione del codice l'autore cita una variabile nell'indirizzo:

3C6 numero di galstar

Si dà il caso che Galstar sia il giocatore, questo sembra essere il numero di vite. Possiamo confermarlo aprendo il monitor e aggiungendo il comando watch w store 3c6. Poi eseguite il gioco, immediatamente il monitor si apre e mostra l'indirizzo del codice che esegue l'inizializzazione della variabile.

`.C:0b48 A9 04 LDA #$04` <- questo è il numero di vite iniziale

```
.C:0b4a 8D C6 03 STA $03C6
```

Se si continua a giocare, alla fine si perde una vita e si trova il codice che diminuisce il numero di vite.

`.C:0dd8 CE C6 03 DEC $03C6` <- questo diminuisce il numero di vite quando viene colpito

Si tratta di una variabile binaria decimale che può essere aumentata fino a 99.

Loader

Ho creato un loader per eseguire il gioco sul mio VIC espanso a +37k utilizzando l'SD2IEC.

Non si avvia immediatamente per dare la possibilità di inserire i POKE. Una volta avviato, premere RETURN per caricare il binario, quindi premere RETURN o spostate il cursore verso il basso per inserire o saltare i POKE.

```
5 poke 55,0:poke56,30:clr:poke648,30:sys58648
10 print"{clr}{7xcrsrd}load 'chr$(34) 'zena2
'chr$(34)', "right$(str$(peek(1
86)),1)",1"
15 print"{4xcrsrd}
poke3618,52:poke3622,13:poke3600,36
20 print"{2xcrsrd}poke6562,7"
25 print"{2xcrsrd}sys2816{15xcrsru}"
```

La riga con i 3 POKE patcha i due tasti e poi l'autofire. La riga successiva modifica il generatore casuale. Non consiglio di patchare le vite perché il gioco diventa troppo facile e noioso.

Risorse

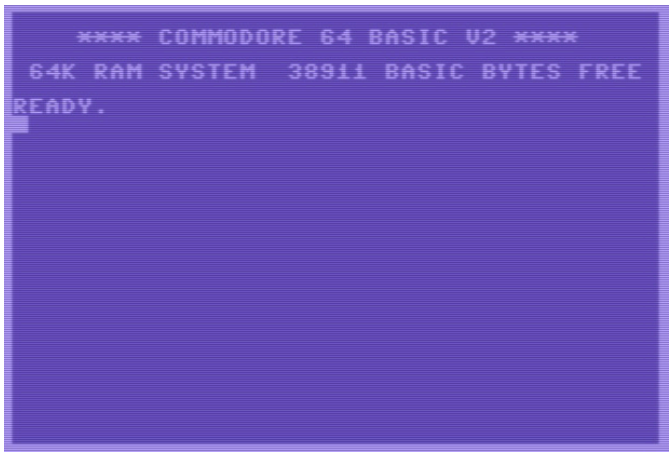
Download del gioco Zena:
- https://drive.google.com/file/d/1AJb8lDOebOGjooisaADCcWc9GafCrJJ9/view?usp=sharing

La rivista PIO numero 11 del 1985:
- https://archive.org/details/pi-o-1985-11/page/136/mode/2up

Commdore 64 - BASIC V2

di Francesco Fiorentini

Qualche tempo fa, in una conversazione con David La Monaca, avevamo espresso la volontá di creare una serie di articoli che provassero a fare luce sulle centinaia di dialetti Basic del Commodore 64. L'impegno sembra improbo, data la notevole mole di lavoro che ci attende e, per alcuni di essi, la pressoché mancanza totale di informazioni. Abbiamo deciso comunque di provarci e... ...non potevamo che iniziare la nostra rassegna se non con il linguaggio che veniva fornito in dotazione con la macchina. Il Commodore 64 infatti, aveva installato in ROM la versione 2.0 del Microsoft Basic.



BASIC, acronimo di Beginner's All-purpose Symbolic Instruction Code, é probabilmente uno dei piú semplici linguaggi di programmazione di terza generazione. Molte delle macchine 8 bit degli anni '80 erano infatti equipaggiate con una versione di questo linguaggio in virtú della sua semplicitá e facilitá di apprendimento.

Il Basic V2, conosciuto anche come Commodore Basic, era stato acquistato da Jack Tramiel dalla Microsoft. Sí, proprio quella, l'azienda di Bill Gates. All'epoca i rapporti di forza delle due societá erano peró nettamente in favore dell'azienda di Tramiel. La Commodore era infatti una solida realtá, mentre la Microsoft era ancora una piccola azienda in costante ricerca di finanziamenti. Emblematica, in questo senso, é la storia dell'acquisto delle licenze del Microsoft Basic. Bill Gates avrebbe voluto vendere il suo Basic dietro pagamento di una royalty di 3 dollari a fronte di ogni installazione, ma Jack Tramiel, forte della sua posizione di vantaggio, rifiutó l'offerta con la famosa frase "Sono giá sposato" e si offrí invece di acquistare il pacchetto pagando soltanto 25000 dollari una tantum per una licenza perpetua. La cifra esatta non é dato saperla, circolano infatti diverse versioni con cifre che oscillano tra i 10000 ed i 30000 dollari. Quello che invece é sicuro, é che Jack Tramiel riuscí a spuntarla su Bill Gates per una cifra decisamente irrisoria, visto poi il numero incredibile di Commodore 64 venduti!

Come dicevamo, il Basic V2 é caricato nella ROM del Commodore 64 ed é immediatemente fruibile all'accensione della macchina. L'interprete viene infatti caricato e messo a disposizione dell'utente che puó cosí cominciare ad utilizzare i comandi Basic in modalitá immediata oppure all'interno di un programma. Si noti che l'utente ha a disposizione del Basic, 38911 bytes di RAM, che corrispondono allo spazio contiguo definito tra $0800-$9FFF (2048-40959).

Comandi del Basic V2

Il Basic V2 consta di 70 comandi:

ABS, AND, ASC, ATN, CHR$, CLOSE, CLR, CMD, CONT, COS, DATA, DEF, DIM, END, EXP, FN, FOR, FRE, GET, GET#, GOSUB, GOTO, IF, INPUT, INPUT#, INT, LEFT$, LEN, LET, LIST, LOAD, LOG, MID$, NEW, NEXT, NOT, ON, OPEN, OR, PEEK, POKE, POS, PRINT, PRINT#, READ, REM, RESTORE, RETURN, RIGHT$, RND, RUN, SAVE, SGN, SIN, SPC, SQR, STATUS/ST, STEP, STOP, STR$, SYS, TAB, TAN, THEN, TIME/TI, TIME$/TI$, TO, USR, VAL, VERIFY, WAIT.

A prima vista 71 comandi potrebbero sembrare molti, ma leggendo attentamente la lista, ci accorgiamo che il Basic V2 é carente di una serie di comandi; specialmente quelli deputati alla gestione della grafica e del sonoro. Tuttavia alcune di queste carenze potevano essere aggirate leggendo o scrivendo rispettivamente in memoria tramite le istruzioni PEEK e POKE: dato che alcuni registri hardware della CPU, del VIC-II e del SID erano mappati direttamente

in memoria, era possibile, usando queste istruzioni, modificare tali registri ed alterare il comportamento del computer.

Purtroppo le carenze del BASIC del Commodore 64 non si limitavano alla mancanza di alcuni comandi, ma erano evidenti anche in termini di velocitá. Ovviamente il confronto con il linguaggio macchina era impietoso, ma anche confrontando il Basic V2 con altri dialetti Basic dello stesso periodo, il Locomotive Basic dell'Amstrad CPC per esempio, il prodotto Microsoft ne usciva con le ossa rotte. Aggiungiamo inoltre la limitazione dei 2 caratteri per i nomi delle variabili ed il delitto é completo.

I nomi delle variabili erano infatti limitati a 2 caratteri di lunghezza, senza peró che nessun errore venisse segnalato al programmatore nel caso un maggior numero di caratteri fosse utilizzato.

Facciamo un esempio:

```
10 PI=10
20 PIPPO=15
30 PINOCCHIO=18
40 PRINT PI
```

Ci aspetteremmo che il risultato di questo programma sia 10. La variabile PI, oggetto della stampa (PRINT) alla riga 40, viene inizializzata con il valore 10 alla riga 10. Invece il risultato del programma é 18, proprio a causa del sopra citato limite dei 2 caratteri per i nomi di variabili.

Attenzione peró perché due variabili aventi lo stesso nome, ma di tipo di dato diverso, rimanevano distinte:

```
10 PI=10
20 PI$="123"
30 PRINT PI
40 PRINT PI$
```

Il risultato di questo programma sará correttamente 10 e "123". Sicuramente i casi riportati negli esempi non possono essere indicati come delle vere e proprio carenze, ma danno un'idea del livello di attenzione necessario a scrivere un programma con il Basic del C64.

IF...THEN

Oggigiorno siamo abituati automaticamente ad associare ELSE al costrutto IF...THEN, ma purtroppo nel Basic V2 non é cosí, essendo questo dialetto mancante dell'operando ELSE. La mancanza dell'ELSE comunque non mina assolutamente l'efficacia del codice, impone peró al programmatore di inserire una serie di controlli aggiuntivi per ovviare a questa lacuna. Soprattutto in caso di 'porting' di programmi da altri dialetti Basic, dobbiamo fare attenzione ai controlli da porre in essere nel caso il dialetto di partenza implementi l'else.

I Token

Per ovvi motivi di ottimizzazione e di spazio, i comandi e le funzioni del Basic V2 non vengono memorizzati come le stringhe che ci appaiono a video. Ognuno di essi é sostituito invece da un codice univoco rappresentato da un valore intero, un byte, chiamato Token.

In fase di esecuzione del codice, il token funge da indice alla jump table o tabella dei vettori che contiene gli indirizzi di memoria delle routine assembly che implementano i comandi BASIC. Al momento di una LIST viceversa, il token viene utilizzato per leggere in quella stessa tabella, il nome per esteso del comando basic da visualizzare a video.

Estensioni Basic

Appare immediatamente chiaro che, volendo creare delle estensioni ai comandi del Basic V2, dovremmo in qualche modo interagire con i token e con la relativa tabella dei vettori. Appare altresí evidente come le limitazioni sinora citate abbiano creato terreno fertile per una decisa proliferazione delle estensioni basic del C64.

Nel corso della lunga vita del gioiellino di casa Commodore, centinaia di programmatori hanno sviluppato decine di queste estensioni e in questo nostro excursus proveremo ad elencarne il maggior numero possibile, cercando di far chiarezza su questo interessante fenomeno ad appannaggio quasi esclusivo del Commodore 64!

Dove reperirlo

La ROM contenente il BASIC V2 viene generalmente fornita insieme agli emulatori del Commodore 64 come Vice, Frodo, CCS64, Hoxs64 o Power64; solo per citarne alcuni dei piú conosciuti.

Riferimenti

"C64 – Guida di riferimento per il programmatore" (1983)
"Guida utente del C64" (1983)
https://pickledlight.blogspot.com/p/commodore-64-guides.html

INFORM 6 - Tutorial 4

di Fredrik Ramsberg - adattato per RMW da Gianluca Girelli

PunyInform Coding 2: Oggetti e azioni

PunyInform è uno strumento per scrivere avventure testuali per computer a 8 bit e computer più recenti. Se hai già installato PunyInform e hai seguito il tutorial PunyInform Coding 1, è giunto il momento di saperne di più su oggetti e azioni. Questo tutorial esamina le diverse parti di un semplice gioco in cui ci sono due puzzle da risolvere, puoi ottenere punti e puoi vincere la partita. Il mio obiettivo è spiegare tutto ciò che non è stato trattato nel tutorial precedente.

Il gioco inizia con il giocatore fuori da una casa, dove pensa che possa esserci il testamento dello zio Billy. Deve passare attraverso la porta principale chiusa a chiave e poi attraverso un'altra porta chiusa a chiave per arrivare alla stanza dove si trova il testamento. Questo è l'intero gioco. Puoi scaricare il codice del gioco e il gioco compilato da questa cartella.

Dichiarazione delle costanti

```
1  !% -~S
2  !% $OMIT_UNUSED_ROUTINES=1
3  !% $ZCODE_LESS_DICT_DATA=1
4
5  Constant Story      "Uncle Billy's Will";
6  Constant Headline   "^An interactive paper hunt.^";
7
8  Constant STATUSLINE_SCORE; Statusline score;
9
10 Constant DEBUG;
11 Constant OPTIONAL_SIMPLE_DOORS;
12 Constant OPTIONAL_NO_DARKNESS;
13 Constant INITIAL_LOCATION_VALUE = Lawn;
14
15 ! Scoring
16 Constant OPTIONAL_SCORED;
17 Constant OPTIONAL_FULL_SCORE;
18 Constant MAX_SCORE 200;
19 Constant OBJECT_SCORE 20;
20 Constant ROOM_SCORE 15;
21
22 Include "globals.h";
23
24 Include "puny.h";
25
```

Game code part 1

Un gioco PunyInform inizia con un set di costanti per indicare il nome del gioco, quali parti opzionali della libreria vuoi usare e altro ancora. In genere devi dichiarare tutte le costanti prima di includere "globals.h".

Alla riga 8 dichiariamo che vogliamo mostrare il punteggio corrente (e il numero di mosse) nella riga di stato. L'altra opzione è mostrare l'ora nella riga di stato, come "3:00 PM". Assicurati di dichiarare sempre che tipo di riga di stato vuoi.

Alla riga 11 diciamo che vogliamo usare il meccanismo Simple Door disponibile in PunyInform. Questo semplifica la creazione di porte.

Per questo gioco, abbiamo deciso che non abbiamo bisogno che la libreria si preoccupi di luce e oscurità: daremo per scontato che ci sia sempre luce presente in tutte le posizioni. Quindi, definiamo OPTIONAL_NO_DARKNESS alla riga 12, per escludere le parti della libreria che gestiscono luce e oscurità. Ciò rende la libreria leggermente più piccola e l'attributo light non verrà definito, poiché non è necessario.

Le righe 15-20 riguardano il punteggio. OPTIONAL_SCORED significa che vogliamo usare l'attributo scoring come un modo semplice per dare punti al giocatore quando raccoglie determinati oggetti o visita determinati luoghi. OPTIONAL_FULL_SCORE significa che vogliamo supportare il verbo "punteggio completo", dando al giocatore un riepilogo dei punti. MAX_SCORE imposta il punteggio massimo. OBJECT_SCORE imposta il numero di punti che il giocatore ottiene per ogni oggetto raccolto che ha l'attributo scoring e ROOM_SCORE il numero di punti per ogni stanza visitata che ha l'attributo scoring. Se non dichiari OPTIONAL_SCORED queste due costanti non hanno effetto.

Quindi includiamo "globals.h" e "puny.h" come facciamo

sempre in un gioco PunyInform. Se dobbiamo dichiarare alcune variabili globali nostre o scrivere routine di punti di ingresso, lo facciamo tra l'inclusione di "globals.h" e "puny.h". Per questo gioco, non abbiamo bisogno di scrivere nulla lì.

Grammatica, verbi e azioni

```
Verb 'knock'
    * 'on' noun -> KnockOn;

Extend 'look'
    * 'under' noun -> LookUnder;

[ KnockOnSub;
    "Knock knock!";
];

[ LookUnderSub;
    "There's probably not much to see there.";
];
```

Game code part 2

PunyInform è dotato di un set standard di verbi di cui la maggior parte dei giochi ha bisogno, come take, drop, open, lock, put, insert ecc. Ogni verbo ha una grammatica, una o più righe che spiegano come può essere utilizzato dal giocatore per formare un comando che porta a un'azione. Righe diverse possono portare a diverse azioni, come "put coat on shelf" che attiva l'azione ##PutOn, ma "put on coat" attiva l'azione ##Wear (le azioni sono (facoltativamente) precedute da ## per farle risaltare nel testo e nel codice).

In questo gioco, vorremmo che il giocatore potesse bussare a una porta e PunyInform non ha un verbo o un'azione per questo. Quindi dichiariamo il verbo "knock". Ogni riga grammaticale inizia con *, aggiunge tutte le preposizioni che il giocatore deve digitare, se presenti, e "noun" per dire che il giocatore deve specificare un oggetto. Puoi anche scrivere preposizioni alternative, come questa: * 'on'/'at'/'in' sostantivo -> KnockOn. E puoi specificare due sostantivi come questo: * 'on' sostantivo 'with' sostantivo -> KnockOn. L'ultima riga grammaticale deve terminare con ; . Scrivere KnockOn dopo -> in una riga grammaticale significa che ora c'è un'azione chiamata ##KnockOn e dovremo creare una routine chiamata KnockOnSub per gestire questa azione, altrimenti otterremo un errore di compilazione. Nota anche che le parole del dizionario (parole che vogliamo confrontare con le parole digitate dal giocatore) devono sempre essere scritte con virgolette singole in Inform.

Vorremmo anche che il giocatore potesse guardare sotto gli oggetti. Questo è un po' diverso, perché PunyInform supporta già il verbo 'look'. Vogliamo solo aggiungere una riga alla grammatica del verbo 'look'. Lo facciamo con Extend. Per questa riga grammaticale, specifichiamo un'altra nuova azione chiamata ##LookUnder. Naturalmente, dovremo anche creare una routine chiamata LookUnderSub per gestire questa azione.

Ora arriviamo alle routine di azione. Una routine di azione definisce cosa fa un'azione di default. Ogni azione rientra in uno dei tre gruppi:

Gruppo 1: Meta-azioni che fanno qualcosa per controllare il gioco stesso, come ##Save (che salva lo stato attuale del gioco in un file) e ##Score (che stampa il punteggio del giocatore).
Gruppo 2: Azioni che cambiano qualcosa nel mondo di gioco o stampano alcune informazioni importanti su di esso, come ##Take (che sposta un oggetto nell'inventario del giocatore) e ##Inv che elenca i beni del giocatore.
Gruppo 3: Azioni che di default non fanno niente se non stampare un messaggio che sostanzialmente si rifiuta di farlo o dice che non ha effetto, come ##Swim, ##Push e ##Yell.
Puoi cambiare cosa fanno le azioni del gruppo 2 e del gruppo 3 per determinati oggetti aggiungendo routine before a questi oggetti o alla stanza in cui avviene l'azione. Puoi anche scegliere di lasciare che qualsiasi altro oggetto presente reagisca usando una routine react_before. Se una di queste routine restituisce true, significa che il comportamento predefinito dell'azione è stato annullato e la routine di azione non verrà nemmeno chiamata.

Per le azioni del gruppo 2, puoi anche usare una routine after nell'oggetto interessato o nella stanza, o una routine react_after in qualsiasi altro oggetto presente per fare qualcosa di speciale e/o stampare un messaggio personalizzato dopo che l'azione è avvenuta. Se la routine restituisce true, la routine di azione non stamperà nulla.

La maggior parte delle azioni che aggiungi a un gioco tendono ad essere azioni del gruppo 3. Entrambe le azioni specificate qui sono azioni del gruppo 3. Ciò significa che la routine di azione dovrebbe solo stampare un messaggio predefinito e basta.

Le locazioni esterne

```
Object Lawn "Lawn"
    with
        description "You are on the lawn. A path leads north to the house.",
        n_to FrontPorch,
        s_to "But you just got here!";

Object FrontPorch "Front Porch"
    with
        description "It's a lavish porch, certainly larger than your entire
            flat.",
        n_to FrontDoor,
        s_to Lawn;

```

Game code part 3

Cominciamo a definire gli oggetti di gioco. Per prima cosa c'è la posizione Lawn. Niente di nuovo. Poi c'è la posizione FrontPorch. L'unico concetto nuovo qui è che un'uscita può portare a un oggetto porta invece che a una posizione. Ne parleremo nella prossima sezione.

Utilizzo di parse_name

```
Object -> DoorMat "door mat"
    with
        parse_name [ w1 w2;
            w1 = NextWord();
            w2 = NextWord();
            if(w1 == 'door' && w2 == 'mat') return 2;
            if(w1 == 'doormat' or 'mat') return 1;
        ],
        before [;
            Take, Remove:
                "That would be incredibly rude.";
            LookUnder, Pull:
                if(parent(MetalKey) == 0) {
                    move MetalKey to location;
                    score = score + 10;
                    "You discover a key!";
                }
        ],
    has supporter enterable;

```

Game code part 4

Ora passiamo allo zerbino, ed è qui che la cosa diventa interessante. Potrebbe sembrare che avremmo potuto semplicemente aggiungere il nome "porta" "zerbino" alle proprietà, ma questo ci metterebbe nei guai. Vedi, c'è una porta d'ingresso in questa posizione (dichiarata più in basso) e questo zerbino. Se il giocatore digita "apri porta" o "esamina porta", non vogliamo che il parser chieda "Quale intendi, la porta d'ingresso o lo zerbino?". Quindi, diamo all'oggetto DoorMat una routine parse_name. Questa è una routine che può leggere le parole dall'input del giocatore usando NextWord() e restituire quante parole in una riga corrispondono a questo oggetto. Questa routine specifica restituisce 2 (il che significa che ha trovato due parole) se il giocatore ha digitato "zerbino" o, in caso contrario, restituisce 1 se il giocatore ha digitato "zerbino" o semplicemente "zerbino". Se il giocatore ha digitato qualsiasi altra cosa, la routine restituisce false (che è uguale a 0), il che significa che non ha trovato nulla. Le routine nelle proprietà, come questa, restituiscono false / 0 per impostazione predefinita, mentre le routine denominate (come le routine di azione nell'ultima sezione) restituiscono true / 1 per impostazione predefinita.

Nella riga 55 la routine inizia con [. Dopo di che possiamo nominare fino a 15 variabili locali che vogliamo usare in questa routine. Queste sono anche usate come parametri, cioè ricevono valori assegnati se qualcuno chiama la routine con uno o più argomenti. Scegliamo di creare due variabili locali denominate w1 e w2. Il ; segna la fine dell'elenco delle variabili locali. Una variabile locale è una variabile che viene creata quando la routine viene chiamata e distrutta quando la routine ritorna. Esiste solo all'interno della routine e non è possibile accedervi in alcun modo da nessun'altra routine. Routine diverse possono avere variabili locali con lo stesso nome, senza rischio di confusione.

Nelle righe 56 e 57 assegniamo valori a queste due variabili. L'assegnazione viene eseguita con un singolo = . Sul lato destro chiamiamo la funzione NextWord e utilizziamo il suo valore di ritorno. Le parentesi significano "chiama la routine con il nome che ho appena dato". Se vuoi inviare argomenti alla routine, li inserisci tra parentesi con virgole tra di loro.

Nella riga 58 vediamo == che viene utilizzato per controllare se un valore è uguale a un altro valore. Vediamo anche l'operatore && che significa AND, quindi l'istruzione if controlla se la prima parola è "door" AND la seconda parola è "mat". Inoltre, Inform ha gli operatori || che significa OR e ~~ che significa NOT.

Nella riga 59 vediamo un altro trucco in Inform: la parola

chiave or. Può essere utilizzata quando si confronta un valore con più altri valori e genera codice più breve e veloce rispetto all'esecuzione dei confronti uno per uno. La riga 59 è funzionalmente equivalente a:

```
if(w1 == 'doormat' || w1 == 'mat') return 1;
```

Nascondere la chiave sotto lo zerbino

C'è una chiave nascosta sotto lo zerbino. Poiché la chiave non può essere vista, in realtà non la mettiamo ancora nella posizione, ma il giocatore dovrebbe essere in grado di trovarla. Quindi, vogliamo che l'oggetto DoorMat sia speciale in due modi:

Se il giocatore guarda sotto lo zerbino o lo sposta da parte, dovrebbe trovare la chiave.

Per evitare di esporre la chiave con una delle azioni standard che molti giocatori provano, come "prendi tutto", vogliamo impedire al giocatore di raccogliere lo zerbino. Lo realizziamo con la routine before. Per prima cosa, reagiamo alle azioni ##Take e ##Remove. In realtà, qui sarebbe bastato solo ##Take, ma ##Take e ##Remove sono le uniche due azioni che devi mai considerare se vuoi impedire a un oggetto di entrare in possesso del giocatore, quindi di solito scrivo entrambe (##Remove viene utilizzato quando un oggetto viene rimosso dall'interno di un contenitore o su un supporto).

Stampa di una risposta e restituzione di true

La sintassi sulle righe 63 e 68 in cui digitiamo semplicemente una stringa racchiusa tra virgolette doppie è una scorciatoia per il comando print_ret, che significa "stampa questo, quindi stampa un carattere di nuova riga e restituisci true". Quindi questi quattro pezzi di codice sono in realtà equivalenti:

```
print "No!";
new_line;
rtrue;
```

```
print "No!^";
rtrue;
```

```
print_ret "No!";
```

```
"No!";
```

Quattro modi per fare la stessa cosa

Guardando sotto il tappetino

Quindi diamo un'occhiata alla seconda parte della routine before. Se l'azione è ##LookUnder o ##Pull, controlliamo se MetalKey ha il genitore 0, il che significa che non è ancora stato inserito nel gioco. Se è vero, spostiamo la chiave nella posizione corrente (location è una variabile globale che contiene l'ID oggetto della posizione corrente), aumentiamo il punteggio di 10 punti, stampiamo un messaggio e restituiamo true. Se la condizione non è soddisfatta, la routine restituirà false e l'azione risponderà con il suo messaggio predefinito. Nota che ogni volta che vogliamo che un'istruzione if esegua più di un comando, dobbiamo racchiudere questi comandi tra { } come facciamo sulla riga 65-69.

Infine, questo oggetto ha due attributi: supporter e enterable. Il primo significa che possono esserci oggetti su questo oggetto. Il secondo significa che il giocatore può anche essere su questo oggetto, usando un comando come "sit on mat" o "enter mat" per salirci.

La chiave

```
73  Object MetalKey "key"
74      with
75          name 'metal' 'key',
76          description "It's a shiny metal key.",
77      has scored;
78
```

Game code part 5

Ora definiamo la chiave. Poiché non c'è -> dopo Object, la chiave non verrà messa nell'ultima posizione definita, ma sarà piuttosto sospesa nel limbo, non dentro o su nulla, quando inizia il gioco. L'unica novità qui è l'attributo scoring. Ciò significa che il giocatore otterrà un certo numero di punti quando la raccoglierà per la prima volta. Il numero di punti è il valore di OBJECT_SCORE , dato alla riga 19.

La porta d'ingresso

```
79  Object FrontDoor "front door"
80      with
81          name 'front' 'door',
82          description "Looks like a very sturdy oak door. Fine craftmanship.",
83          when_open [;
84              if(location == FrontPorch)
85                  "The front door is open, beckoning you inside.";
86              else
87                  "The front door is open, showing you a glimpse of the porch
88                      and lawn outside.";
89          ],
90          when_closed "The sturdy front door is closed.",
91          before [;
92              KnockOn:
93                  "You knock and wait in silence. There is no reply.";
94          ],
95          found_in FrontPorch Entrance,
96          door_dir (n_to) (s_to),
97          with_key MetalKey,
98      has static door openable lockable locked;
99
```

Game code part 6

Ora è il momento di definire la nostra prima porta. Utilizzeremo il meccanismo Simple Door di PunyInform, poiché semplifica il codice e il più delle volte soddisfa le esigenze. Una porta in PunyInform è qualsiasi oggetto che abbia l'attributo door. Una porta è un passaggio tra posizioni. In genere esiste in due posizioni e fornisce un mezzo per andare dalla prima posizione alla seconda o dalla seconda posizione alla prima.

when_open e when_closed svolgono lo stesso ruolo di initial ma solo per porte e contenitori. Vale a dire, l'oggetto ottiene un paragrafo a sé stante nella descrizione della stanza. Se l'oggetto è aperto, viene utilizzato when_open, altrimenti viene utilizzato when_closed. In questo caso, vogliamo che il testo when_open sia diverso a seconda che stiamo vedendo la porta dall'esterno o dall'interno della casa, quindi utilizziamo una routine. Per when_closed ci basta una stringa statica.

Toc toc, chi è? Ora arriva il motivo per cui abbiamo creato il verbo "knock" e l'azione ##KnockOn in precedenza: vogliamo dare una risposta appropriata se il giocatore prova a bussare alla porta d'ingresso, poiché sembra una cosa abbastanza sensata da fare. Lo facciamo con una routine before. Se l'azione è ##KnockOn , stampiamo un messaggio e restituiamo true, impedendo così la stampa del messaggio predefinito.

Oggetti mobili

In qualsiasi momento, ogni oggetto può trovarsi solo in una posizione, ma PunyInform ha un meccanismo per spostare gli oggetti ogni volta che il giocatore si muove, quindi un oggetto può sembrare che esista in più posizioni. Questo è talvolta definito come oggetti mobili ed è controllato con la proprietà found_in. Il valore di found_in dovrebbe essere una routine che restituisce true se l'oggetto dovrebbe trovarsi nella posizione corrente, oppure un elenco di posizioni in cui dovrebbe trovarsi (noterai che un elenco di valori in Inform è anche definito array nella documentazione).

Se vogliamo usare il meccanismo Simple Door per una porta, quella porta deve avere una proprietà found_in e il suo valore deve essere un array con esattamente due valori. In questo caso, posizioniamo la porta in FrontPorch e Entrance.

Far funzionare la porta

Una porta deve avere una proprietà door_dir, per indicare in quale direzione si trova una porta. Per una porta normale, questo dovrebbe essere un singolo valore o una routine e di solito devi scrivere una routine che restituisce valori diversi a seconda della posizione corrente. Se usi il meccanismo Simple Door e la porta ha un array con due valori in found_in , puoi semplicemente fornire un array con due valori qui. Se il giocatore si trova nella prima posizione nell'array found_in, viene utilizzato il primo valore, altrimenti viene utilizzato il secondo valore. Utilizziamo le parentesi attorno ai valori per evitare un avviso dal compilatore.

Per una porta normale, dobbiamo anche fornire una proprietà door_to, che indica dove conduce la porta. Anche questo dovrebbe essere un singolo valore o una routine. Per una Simple Door, possiamo semplicemente saltarlo. Se il giocatore si trova nella prima posizione nell'array found_in, la porta conduce alla seconda posizione nell'array e viceversa.

Per un oggetto che è bloccabile, la proprietà with_key viene utilizzata per dire quale chiave si adatta alla serratura. Poi arrivano gli attributi. Usiamo static per dire che questo oggetto è fisso in un posto e non può essere raccolto dal giocatore. Usiamo door per dire che è una porta. Dobbiamo aggiungere openable per dire che può essere aperto e chiuso e lockable per dire che può essere bloccato e sbloccato. Infine aggiungiamo locked per dire che la porta è bloccata dall'inizio. Ed ecco fatto, abbiamo una porta completa!

Aggiunta di un contenitore accessibile

```
Object Entrance "Entrance"
    with
        description "This area feels warm and welcoming, with its wooden panels
            and many potted plants. A small sign next to the doorway to the
            east tells you there's an office in that direction. Below the sign
            are two boxes on the wall, one blue and one green.",
        s_to FrontDoor,
        e_to OfficeDoor,
    has scored;

Object -> GlassBox "huge glass box"
    with
        name 'huge' 'glass' 'box' 'boxes//p',
        inside_description "The box seems surprisingly roomy now that you're in
            it. Of course the glass sides help - you'd feel a lot more
            constrained if the sides weren't transparent.",
        before [;
            Take, Remove, Turn, Push, Pull:
                "It weighs a ton at least!";
        ],
    has static container openable enterable transparent;

```

Game code part 7

Ecco la definizione di Entrance, la stanza all'interno di FrontDoor. Puoi anche vedere che un'uscita conduce a FrontDoor e un'altra conduce a un'altra porta chiamata OfficeDoor. Questa posizione ha l'attributo scoring, il che significa che il giocatore ottiene un certo numero di punti quando arriva qui per la prima volta. Il numero di punti è il valore di ROOM_SCORE , dato alla riga 20.

Ci sono diversi oggetti interessanti in questa posizione. Iniziamo con la scatola di vetro. Come vedrai presto, ci sono diverse scatole in questa posizione. Ha senso consentire al giocatore di fare riferimento a tutte contemporaneamente, come in "prendi le scatole". Per supportare ciò, aggiungiamo la parola "scatole" alla proprietà name, solo che aggiungiamo il flag plurale, per dire che è la forma plurale del nome, in questo modo: "scatole//p".

Inoltre, stiamo rendendo questa scatola accessibile e vogliamo aggiungere una descrizione per quando il giocatore è all'interno della scatola. Lo facciamo con inside_description. Questo può essere utilizzato anche per i sostenitori accessibili.

Se il giocatore prova a spostare la scatola, vogliamo dirgli che è molto pesante, quindi aggiungiamo una routine before che cattura le azioni comuni utilizzate per spostare le cose.

Procediamo con gli attributi. Questo è un oggetto statico, il che significa che qualsiasi tentativo di prendere o spostare l'oggetto ottiene una risposta come "È fissato in posizione". (solo che abbiamo scelto di ignorare tutti o la maggior parte di essi nella routine before). È un contenitore, il che significa che altri oggetti possono essere al suo interno. openable significa che può essere aperto e chiuso. enterable significa che il giocatore può entrare nell'oggetto. transparent significa che è trasparente, il che significa che possiamo guardare il contenuto anche quando è chiuso. transparent significa anche che se qualcosa emette luce all'esterno, c'è luce anche all'interno e viceversa, ma questo non ha importanza in questo particolare gioco, poiché abbiamo definito OPTIONAL_NO_DARKNESS alla riga 12.

La collana

```
122 Object -> -> Necklace "necklace"
123     with
124         name 'necklace' 'string',
125         description "It consists of a piece of string with an emerald, a ruby
126             and a sapphire on it.",
127         before [ loc;
128             Attack:
129                 if (ObjectIsUntouchable(noun))
130                     rtrue;
131                 loc = parent(self);
132                 remove self;
133                 move Emerald to loc;
134                 move Ruby to loc;
135                 move Sapphire to loc;
136                 score = score + 20;
137                 "You break the necklace, leaving you with three gems.";
138         ],
139     has transparent;
140
```

Game code part 8

Ora aggiungiamo una collana. Ha due frecce (->) dopo Object, il che significa che si trova nell'ultimo oggetto definito con una singola freccia, in questo caso la scatola di vetro.

Il giocatore dovrebbe essere in grado di rompere la collana per liberare le gemme. Lo facciamo con una routine before. Avremo bisogno di una variabile locale e la chiameremo loc (abbreviazione di "location of the necklace"). Catturiamo l'azione ##Attack, che è ciò che otteniamo se il giocatore digita "break necklace", "destroy necklace" o "wreck necklace". Per prima cosa utilizziamo la routine PunyInform ObjectIsUntouchable per verificare se la collana è intoccabile dal giocatore. In tal caso, ObjectIsUntouchable stampa un messaggio che spiega al giocatore perché non può raggiungere la collana e restituiamo true per dire a PunyInform di non fare o stampare altro. Nota che qui utilizziamo noun, che è sempre impostato sull'oggetto primario dell'azione corrente. Quando siamo in una routine before (almeno una che non appartiene a una stanza), noun ha lo stesso valore di self, quindi possiamo usarli in modo intercambiabile. Poiché abbiamo deciso che il giocatore può raggiungere la collana, eseguiamo una sequenza di comandi:

Impostiamo loc sul genitore corrente della collana. Potrebbe essere la scatola di vetro in cui l'abbiamo trovata, ma potrebbe anche essere il giocatore, o potrebbe essere la posizione dell'ingresso ecc.
Quindi rimuoviamo la collana, ovvero la rimuoviamo dall'albero degli oggetti e la mettiamo nel vuoto dove il giocatore non la vedrà mai.
Successivamente spostiamo le tre gemme che fanno parte

della collana nella posizione in cui si trovava la collana. Aumentiamo il punteggio

Stampiamo un messaggio e restituiamo true, segnalando che PunyInform non dovrebbe stampare il messaggio predefinito per ##Attack.

La collana ha un singolo attributo: transparent . Questo serve a segnalare che se ci sono oggetti in questo oggetto, possono essere visti dal giocatore. Posizionare oggetti all'interno di altri oggetti significa cose diverse a seconda del tipo di oggetto dell'oggetto padre:

Per un contenitore (una borsa, una scatola, un'auto ecc.), significa che gli oggetti al suo interno sono in effetti all'interno del contenitore.

Per un supporto (un letto, un tavolo, un tappeto ecc.), significa che gli oggetti al suo interno sono sul supporto.

Per un oggetto animato (una persona, un animale, un robot ecc.), significa che gli oggetti al suo interno sono tenuti dall'oggetto animato.

Per tutti gli altri oggetti, significa che gli oggetti sono parte dell'oggetto, come i pulsanti sono parte di un registratore di cassa ecc.

L'ultimo caso è quello di cui ci stiamo occupando qui. Guarda il prossimo pezzo di codice per vedere le gemme che fanno parte della collana.

Le gemme

```
Object -> -> -> Emerald "emerald"
    with
        name 'emerald' 'gem' 'gems//p',
        article "an";

Object -> -> -> Ruby "ruby"
    with
        name 'ruby' 'gem' 'gems//p';

Object -> -> -> Sapphire "sapphire"
    with
        name 'sapphire' 'gem' 'gems//p';

```

Game code part 9

Aggiungiamo ora le tre gemme. Le tre frecce indicano che sono nella (cioè fanno parte della) collana che abbiamo appena definito. Tutte e tre hanno la parola plurale "gemme" aggiunta alla loro proprietà nome, così il giocatore può "prendere gemme". Lo smeraldo ha bisogno dell'articolo indeterminativo "un" e quindi gli diamo la proprietà articolo e l'articolo desiderato come stringa.

Mettere le gemme nella collana in questo modo significa che vanno ovunque vada la collana, il giocatore può esaminare le gemme, ma se prova a prenderne una riceve un messaggio che fa parte della collana.

La serratura della porta dell'ufficio

```
Object -> BlueBox "blue box"
    with
        name 'blue' 'box' 'boxes//p',
        description "It's a small metal box, firmly attached to the wall next to
            the office door.",
        describe [; rtrue; ],
    has static container open;

Object -> GreenBox "green box"
    with
        name 'green' 'box' 'boxes//p',
        description "It's a small metal box, firmly attached to the wall next to
            the office door.",
        describe [; rtrue; ],
    has static container open;

```

Game code part 10

Oltre alla scatola di vetro con la collana, ci sono anche una scatola blu e una scatola verde, attaccate al muro accanto alla porta dell'ufficio. Queste sono in effetti parte del meccanismo di serratura della porta dell'ufficio. Se metti uno zaffiro nella scatola blu e uno smeraldo in quella verde, la porta si sblocca. Il codice per farlo accadere, tuttavia, non è nelle scatole ma nella porta stessa.

Il codice per definire le scatole non riserva molte sorprese. L'unica cosa nuova qui è la proprietà describe. Questa viene utilizzata quando potresti voler stampare un paragrafo di testo per un oggetto nella descrizione della stanza. Se lo fai, stampi una nuova riga, qualsiasi testo tu voglia, un'altra nuova riga e poi restituisci true. Se decidi di non voler stampare un paragrafo separato per questo oggetto, restituisci false. Ma c'è un trucco che puoi fare qui: non stampare nulla ma restituisci true. Questo impedirà a "look" di stampare qualsiasi cosa sull'oggetto. Le scatole sono menzionate nella proprietà descrizione della stanza, quindi non vogliamo che il gioco le mostri di nuovo.

La porta dell'ufficio

(vedi fig12, pagina seguente)

Il giocatore può posizionare vari oggetti nelle caselle blu e verdi. Ogni volta che il giocatore ha inserito qualcosa in qualcos'altro ("metti smeraldo nella casella verde"

```
Object OfficeDoor "office door"
    with
        name 'office' 'door',
        react_after [;
            Insert, Take, Remove:
                if(Sapphire in BlueBox && children(BlueBox) == 1 &&
                        Emerald in GreenBox && children(GreenBox) == 1 &&
                        self has locked) {
                    give self ~locked;
                    score = score + 70;
                    "A loud click is heard from the office door.";
                }
        ],
        describe [; rtrue; ],
        found_in Entrance Office,
        door_dir (e_to) (w_to),
    has static door openable lockable locked;
```

Fig. 12 - Game code part 11

innesca un'azione ##Inserisci) o ha raccolto qualcosa, vogliamo controllare se sono soddisfatte le condizioni per sbloccare la porta. Quando vogliamo che un oggetto (in questo caso la porta dell'ufficio) reagisca a un'azione che avviene nelle vicinanze ma in cui questo oggetto potrebbe non essere direttamente coinvolto, possiamo usare react_before e react_after. Funzionano in modo molto simile a before e after. Possiamo usare noun e second per controllare quali nomi primari e secondari sono usati nell'azione. In questo caso, però, non ne abbiamo bisogno. Controlliamo se sono soddisfatte le seguenti condizioni:

Lo zaffiro è nella scatola blu
C'è esattamente un oggetto nella scatola blu
Lo smeraldo è nella scatola verde
C'è esattamente un oggetto nella scatola verde
La porta è chiusa a chiave

Se tutte queste condizioni sono soddisfatte, noi:

Sblocchiamo la porta
Diamo al giocatore alcuni punti
Stampiamo un messaggio e restituiamo true, il che significa che PunyInform non deve stampare nulla
Lo sblocco della porta avviene impostando l'attributo bloccato a 0 per l'oggetto OfficeDoor. Un attributo è un valore di bit, può essere 0 o 1. Ecco cosa puoi fare con un attributo:

Puoi controllare se l'attributo bloccato è 1 per l'oggetto X con if(X ha bloccato)
Puoi controllare se bloccato è 0 con if(X non ha bloccato)
Puoi impostare bloccato a 1 con give X bloccato
Puoi impostare bloccato a 0 con give X ~locked

Il resto della definizione di questo oggetto è molto simile alla porta d'ingresso. Una differenza è che, sebbene abbia l'attributo lockable, non ha la proprietà with_key. Ciò significa che il giocatore può dire che la porta è chiusa a chiave, ma non c'è una chiave che possa sbloccarla usando il verbo "unlock".

La posizione finale

```
Object Office "Office"
    with
        name 'desk' 'chair',
        description "It's a neat and tidy little office. There's a desk and a
            chair near the window.",
        w_to OfficeDoor,
    has scored;

Object -> Will "Will"
    with
        name 'will' 'piece' 'of' 'paper',
        initial "There's a piece of paper on the desk.",
        description [;
            deadflag = GS_WIN;
            score = score + 50;
            "It's uncle Billy's Will, and it says he leaves everything to you!
                You are now rich!";
        ];
```

Game code part 12

Siamo arrivati alla posizione finale, l'ufficio. Descriviamo una scrivania e una sedia, ma in realtà non sono così importanti, quindi preferiremmo non implementarle come oggetti. Invece mettiamo queste parole nella proprietà name della posizione, così il giocatore riceve un messaggio che dice che non ha bisogno di fare riferimento a questi oggetti, se prova a interagire con loro.

Il testamento ha la proprietà iniziale, il che significa che questa stringa verrà stampata come un paragrafo a sé stante quando il giocatore visualizza questa stanza, finché l'oggetto non si è spostato. Se il giocatore prende il testamento e lo lascia cadere di nuovo, verrà semplicemente elencato insieme a tutti gli altri oggetti normali nella posizione.

E poi c'è la descrizione, ovviamente. Se il giocatore prova a esaminare o leggere il testamento, viene consultata la proprietà description. Se è una routine, viene eseguita. Quindi questo è un buon punto per noi per concludere il gioco. Ecco cosa facciamo:

Impostiamo deadflag su GS_WIN (che è 2, ma è più facile ricordare GS_WIN) ("GS" sta per Game State, a proposito, poiché questa variabile controlla se la partita continua o finisce). Se invece lo imposti su GS_DEAD, la partita finisce

anche questa, ma con un messaggio che dice che il giocatore ha perso.

Aumentiamo il punteggio.

Stampiamo la descrizione del testamento. (Solo scrivendo una stringa tra virgolette doppie in questo modo, stampiamo anche una nuova riga, il che è positivo, e restituiamo true, il che non importa: la proprietà description dovrebbe sempre stampare una descrizione e il suo valore di ritorno viene semplicemente ignorato.)

Dopo che l'azione è stata completata, PunyInform nota che deadflag è ora GS_WIN e termina la partita.

La routine Initialise

```
[Initialise;
    print "^^Ever since your dear uncle Billy got sick, his spoilt kids have
        been fighting over the inheritance. Billy told you in confidence that
        he's been working on a will to fix that problem, but you don't know
        if he managed to finish it before he passed away last month. You've
        worked hard to find it, but this far you've failed. This morning you
        got news from Billy's lawyer that Billy owned another house that noone
        knew about. You realize you're clutching at straws here, but you have
        to check it out. The cab pulls away and you're on your own.^^";
];
```

Game code part 13

Infine, è il momento di scrivere la routine obbligatoria Initialise, che viene chiamata quando inizia il gioco. Possiamo usare questa routine per spostare le cose, avviare i timer ecc., ma per questo gioco tutto ciò di cui abbiamo bisogno è stampare un'introduzione al gioco.

Il gioco finito

Dopo aver compilato il gioco in un file Z-code, puoi giocarci con l'interprete Z-code che preferisci. Lo screenshot qui sopra è di Windows Frotz.

Link

Il codice sorgente visto in questo tutorial, così come il gioco compilato (demogame_2.z3) possono essere scaricati da questa cartella:

https://drive.google.com/drive/folders/1SIqRNqi99LunTq9jg2-4sWzVGSVhhsg6?usp=sharing

File View Help

```
Front Porch                                   Score: 10       Moves: 3
Ever since your dear uncle Billy got sick, his spoilt kids have been fighting over the
inheritance. Billy told you in confidence that he's been working on a will to fix that
problem, but you don't know if he managed to finish it before he passed away last
month. You've worked hard to find it, but this far you've failed. This morning you got
news from Billy's lawyer that Billy owned another house that noone knew about. You
realize you're clutching at straws here, but you have to check it out. The cab pulls
away and you're on your own.


Uncle Billy's Will
An interactive paper hunt.
Release 1 / Serial number 230919 / Inform v6.41 PunyInform v5.0 DR

Lawn
You are on the lawn. A path leads north to the house.

> n
Front Porch
It's a lavish porch, certainly larger than your entire flat.

The sturdy front door is closed.

You can also see a door mat here.

> knock on door
You knock and wait in silence. There is no reply.

> look under the door mat
You discover a key!

[The score has just gone up by 10 points.]
```

Fig. 15 - Il gioco in azione

Fino ad arrivare laddove nessun SID è mai giunto prima...

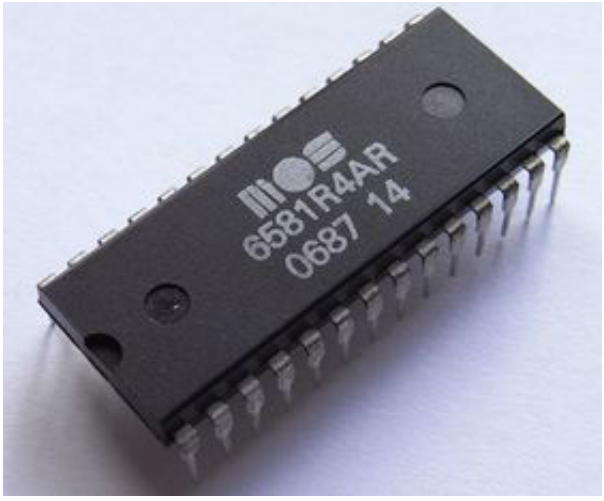
STUDIO DI METODI DI CAMPIONAMENTO AUDIO PER LA RIPRODUZIONE DI BRANI DI ALTA QUALITÀ SU COMMODORE 64

di Marco Pistorio

Eccoci ad un articolo che alcuni di Voi aspettavano da un po' di tempo, grossomodo da quando ho iniziato a pubblicare su FB e sul mio canale youtube (https://www.youtube.com/@mpsoftita-retroprogramming8233) spezzoni di brani audio campionati e poi riprodotti tramite il C64.
Qualcuno gridava al miracolo, qualcun altro diceva che si trattava di montaggi, altri ancora dicevano che via emulatore era tutto facile (ma ho fatto dei video mostrando i risultati ottenuti anche su macchina reale) ed infine, altri ancora dicevano che non era possibile ottenere quella resa audio dal buon vecchio SID.

Il preambolo importante, a mio avviso, è che "l'eroe" non sono io, non è il player bensì è il buon vecchio SID, il chip dedicato al comparto audio di cui fu equipaggiato il nostro amato Commodore 64.
Questo integrato fu costruito con lungimiranza tanto che ancora oggi abbiamo difficoltà a comprendere appieno come faccia a riprodurre questa grande varietà di suoni, con un livello di qualità di tutto rispetto, nonostante il fatto che venne concepito nell'ormai lontano 1981.
Un plauso va innanzitutto a chi lo realizzò, ovvero all'ingegnere Robert Yannes.
A tal proposito scambiai un paio di chiacchiere con l'amico FlavioWeb Pasqualin, che contestualmente saluto.
Anche lui si trovava concorde con me circa la lungimiranza di chi ha realizzato questo gioiellino dell'elettronica ormai oltre quarant'anni fa.

A questo punto, ci sono alcune cose da mettere in chiaro.
Innanzitutto, il campionamento NON AVVIENE in tempo reale sul C64 bensì trattasi di spezzoni di brani campionati da PC.
Più precisamente, ho adoperato il software "Audacity", sia per il campionamento che per l'esportazione di tali campioni in un formato .raw
E qui nasce un secondo grosso guaio.
Serve tanta RAM per contenere i campioni dei brani.
Utilizzando una frequenza di campionamento di 14,4 KHz, occorrono, per ogni secondo di audio, 14400 campioni ovvero, in termini di RAM occupata circa 14 Kb(!).



MOS Technology SID

In altre parole, con 3 o 4 secondi di brano mi gioco tutta la RAM a disposizione del Commodore 64.
Il mio obiettivo era arrivare a riprodurre almeno 1 minuto per brano, e mi servirebbero pertanto circa 1 Mb di RAM, cioè 16 volte la memoria a disposizione del Commodore 64.
E quindi ho provato a fare mente locale, per trovare una strada percorribile.
La compressione non andava bene per due motivi:
il primo è che bisogna decomprimere i dati compressi, e ciò richiede preziosissimo tempo macchina.
il secondo è che, anche sfruttando gli algoritmi più performanti, non avrei potuto comunque conservare 1 Mb di campioni in 64 Kb di RAM (né in meno spazio, ovviamente)
Scartando la compressione...come trovare 1 Mb di RAM per il C64?
Mi è venuto in mente che la cartuccia EasyFlash dispone proprio di 1 Mb di RAM da poter sfruttare a nostro piacimento.
E' vero, si tratta di un hardware che fu messo in commercio a partire dal 2009.
Tuttavia, con l'aggiunta di questa RAM, posso utilizzare il C64 ed in particolare il chip 6510 (CPU) ed il 6581/8580 (SID), esattamente per come sono stati fabbricati, arrivando all'obiettivo che mi sono prefisso senza troppe difficoltà.
Ecco un secondo spunto di riflessione importante.

Riusciamo a renderci conto meglio delle potenzialità del SID ai giorni nostri grazie anche a questi nuovi strumenti

di cui possiamo disporre!

Mi ero già interessato, nel corso degli anni, alle tecniche di riproduzione audio che coinvolgevano il C64 ed in particolare il SID.
Già nei primissimi anni dell'arrivo sul mercato del Commodore 64 alcune di queste peculiarità erano discretamente sfruttate.
Ricordo ad esempio il vocale "GHOSTBUSTERS" nella intro del famoso gioco rilasciato dalla Activision nel 1984 per il nostro biscottone, ma non è il solo esempio.
Anche nel gioco "Impossible Mission", anche lui del 1984, viene sfruttata questa funzionalità della sintesi vocale.
Celebre, per gli appassionati, la frase "Another visitor. Stay a while... stay forever!" che viene recitata nel gioco.
Non parliamo poi del S.A.M. (Software Automatic Mouth) rilasciato nel 1982, un software che riproduceva in maniera stupefacente (per quegli anni) del testo scritto in lingua inglese.
Questi risultati si riescono ad ottenere sfruttando il DAC (Digital-Analog Converter) che è accessibile agendo sul registro che gestisce il volume del SID.
Tali risultati ottenibili di norma sono tuttavia modesti, in quanto per tale conversione sono disponibili solo 4 bit.
Si riesce così ad ottenere una discreta risoluzione per simulare un contenuto vocale, ma non si può andare molto oltre, a meno di impiegare altre strategie insieme all'impiego del registro del volume del SID.

Un risultato di alta qualità, a titolo di esempio, è stato ottenuto dal grande Antonio Savona (che saluto da queste queste righe, nel contempo) che nel 2017 pubblicò un suo player.
Questo player sfrutta il registro del volume del SID, insieme a tabelle di lookup per migliorare la percezione dell'audio che viene riprodotto.
Anche lui adottò lo stesso supporto che ho scelto io per contenere i dati gestiti dal suo player, ovvero una cartuccia EasyFlash.
Tramite una pre-elaborazione (che include una tecnica di quantizzazione vettoriale e tecniche statistiche raffinate tipo K-means) del segnale da processare, partendo dal singolo campione da riprodurre si arriva al valore, calcolato opportunamente e prelevato da una tabella di lookup, valore che viene scritto sul registro del volume del SID e che viene quindi riprodotto.
Grazie ad una sapiente gestione dei cicli macchina, il suo player raggiunge una frequenza di campionamento di 48 Khz!

Per chi volesse approfondire:
- https://csdb.dk/release/?id=162941
- https://csdb.dk/release/?id=162951
- http://brokenbytes.blogspot.com/

Ma veniamo al metodo che invece ho deciso di adoperare io.
Personalmente ho preferito evitare le pre-elaborazioni perché mi sembrava un po' come giocare con un mazzo di carte 'truccate'.
La pre-elaborazione sposta il carico della elaborazione dei dati sul PC, facilitando i compiti del C64, e del SID in particolare.
Strategia giusta e legittima, guardando ai risultati ottenibili, ma la mia idea era di impiegare un metodo più veloce, più semplice e che non richiedesse una eccessiva "processazione" del dato.
Ho preferito quindi un campionamento che sfruttasse 8 bit anziché solo 4 bit, per garantirmi una buona sonorità.

Qui mi sono avvalso degli studi pubblicati su un paio di articoli pubblicati sui numeri 20 e 21 di C=Hacking, che risalgono agli anni 2001-2002.
Più precisamente, in questi articoli si studiava la possibilità di adoperare la modulazione a larghezza di impulso (PWM) per riprodurre audio campionato con una buona qualità.
Alcuni esempi di codice riguardavano C64 con SuperCPU ma c'era qualcosa che avrebbe potuto girare su un c64 "breadbin" senza alcuna aggiunta.
Accontentandosi della esigua quantità di RAM a disposizione del C64 non si arrivava a riprodurre più di 3 o 4 secondi di brano, naturalmente.
Ma questo problema si aggira con l'impiego della EasyFlash.
Tale scelta ha comportato una modifica del codice inizialmente trovato e che avevo deciso di impiegare poichè dovevo integrarlo con la gestione dei banchi di RAM della cartuccia.
Questa gestione infatti richiede cicli macchina aggiuntivi (leggasi anche 'ritardo') e senza correttivi il ritardo introdotto generava un "tick" di sottofondo piuttosto fastidioso.

Il codice dal quale sono originariamente partito, per chi fosse curioso, è questo che segue:

```
//////////////////////////////////////////////////////////////////////////
; 7.5 bit sample player for C64
; (C) 2001 Levente H rsfalvi

; Ideas obtained from:
;
```

```
; - Steve Judd  (C= Hacking 18)
; - Dag Lem (reSID emulator engine)

; Runs on a stock C64 (PAL or NTSC), and plays an unsigned 8-bit sample
; using PWM.

; Sampling rate: 68 clock cycles
; 17734470 / 18 / 68 = 14488.946078 Hz for PAL
; 14318180 / 14 / 68 = 15040.105042 Hz for NTSC

; Table Driven Assembler + Comlink
; (probably easy to compile using another 6502 cross assemblers too)

; This routine is 'free software', I just don't bother quoting GPL


#ifndef LO
#define LO(x) (x&$ff)
#define HI(x) (x>>8)
#endif

         *= $0900

        *= *-2
        .word *+2                       ;Load address for the binary

        sei
        lda #$0b
        sta $d011                       ;blank screen
        lda #$35
        sta $01                         ;ROMs off

        lda #$0f                        ;Volume + filter
        sta $d418
;        lda #$07                        ;Lowpass filter (but it sounded so
;        sta $d417                       ;bad on my 6581R4 that I commented
;        lda #$a0                        ;it out
;        sta $d416

        lda #$00                        ;Set SID freq registers
        sta $d400
        sta $d407
        sta $d40e
        lda #$40                        ;$4000 --> the wave accu is
        sta $d401                       ;incremented by $004 each CPU clock
        sta $d408                       ;cycles
        sta $d40f
        lda #$f0                        ;maximum sustain level for all chns
        sta $d406
        sta $d40d
        sta $d414

        lda #LO($0a00)                  ;start address of sample in memory
        sta ptr
        lda #HI($0a00)
        sta ptr+1
        lda #$cd                        ;end address high. Zpg ($fb) is used
        sta $fb                         ;because of timing reasons

        ldx #$49                        ;pre-load values for the SID command
        ldy #$41                        ;registers...

ptr     =*+1
loop    lda $face    ;4                 ;Load next sample...
        sta $d402    ;8                 ;put to the sample width reg LO
        stx $d404    ;12                ;RESET SID voice
        sty $d404    ;16                ;...then start wave counter
        adc #$01     ;18                ;Increase sample value
        sta $d409    ;22                ;(presumably, there'll never be a
        stx $d40b    ;26                ;carry), then go on for next SID
        sty $d40b    ;30                ;channels.
        adc #$01     ;32
        sta $d410    ;36
        stx $d412    ;40
        sty $d412    ;44
        inc ptr      ;50                ;Increase sample pointer
        bne loop1    ;52  (when jumps, 53)
        inc ptr+1    ;58
        lda ptr+1    ;62
        cmp $fb      ;65
        bne loop     ;68

        lda #LO($0a00)                  ;Reload it (timing is broken, but no
        sta ptr                         ;matter since it only happens per one
        lda #HI($0a00)                  ;full sample replay)
        sta ptr+1
        jmp loop

loop1   bit $face    ;57                ;timing...
        bit $face    ;61
        bit $face    ;65
        jmp loop     ;68                ;Then loop.


i_sample =$0a00                         ;Just to include the sample binary
                                        ;(TASM + Comlink feature)
        .export i_sample

.end
```

//

I crediti relativi al programma sono già indicati nel sorgente. Possiamo descrivere l'algoritmo proposto così:

- Init: volume al massimo, livello di sustain della voce 1 al massimo, inizia l'attacco. La frequenza viene impostata (=$4000), quindi il contatore delle onde viene incrementato di 4 ogni ciclo di clock del processore.

Loop:
- carica il valore del campione successivo e inseriscilo nel registro basso della larghezza di impulso ($d402; assicurati che $d403 sia 0).
- Imposta il bit di test e cancella il bit di test (reset del contatore). (Trattasi dei due valori precalcolati $49 e $41 che vengono memorizzati inizialmente rispettivamente all'interno dei registri X ed Y N.d.R)
- Aumenta il puntatore del campione, un po' di ritardo, quindi esegui il loop. Il ritardo deve essere di 64 cicli di clock + il tempo in cui il bit di test viene mantenuto impostato (4 cicli per ciascun opcode, se si utilizza STX $d404: STY $d404 immediatamente con valori precaricati).

Cosa accadrà? Il valore del campione a 8 bit viene inserito direttamente nel registro della larghezza di impulso (gli MSB del registro della larghezza di impulso vengono cancellati!...). Il contatore di onde viene avviato (bit di test rilasciato) e aumenta di 4 per ogni ciclo della CPU (=conta 256 in 64 cicli).
Dopo un po' di tempo, il contatore raggiungerà il valore nel registro della larghezza di impulso. Ciò avviene

esattamente dopo (valore campione a 8 bit /4) cicli, per quanto sopra. In questo ciclo (o nel prossimo?...) il SID trasforma la sua uscita di impulso in 0. Voilà!

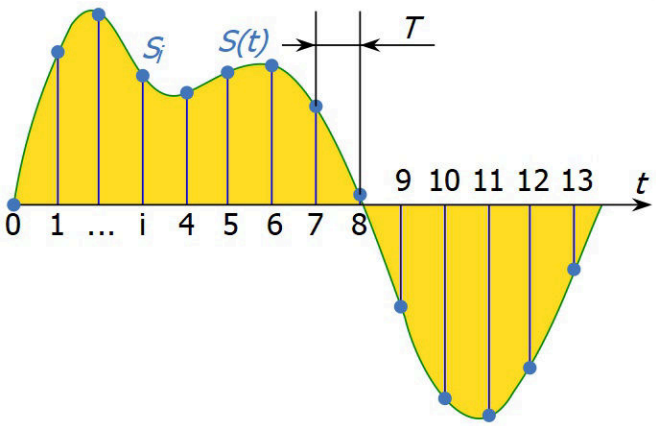
Questa è la spiegazione che ha fornito proprio l'autore di questo sorgente.

E' importante sottolineare il ruolo fondamentale dei due valori precaricati nei registri X ed Y e poi memorizzati successivamente nel registri di controllo di ciascuna voce del SID.

All'interno di questi miei commenti viene spiegato meglio il significato di questi valori $49 e $41.

```
// pre-load values for the SID command
// registers...

//  NOISE  PULSE SAW  TRI  TEST  RING SYNC GATE
//   0      1     0    0    1     0    0    1

// #$49


//  NOISE  PULSE SAW  TRI  TEST  RING SYNC GATE
//    0     1     0    0    0     0    0    1

// #$41

/* Playing a sound:

1. Set frequency and ADSR, and in case you use the pulse wave also
set the pulse width.
2. Set the waveform and the GATE bit in the control register.
Setting the gate bit will start the ADSR envelope generator.
3. Wait for as long as you wish the note to be played.
4. Clear gate bit. This will start the release phase of the
ADSR.
(https://www.oxyron.de/html/registers_sid.html)
*/
```

Non viene evidenziato in questi commenti il ruolo del segnale di TEST. Il segnale di TEST, così come già accennato, permette di suonare il campione ed udirlo per un ben preciso periodo di tempo. Superato questo periodo di tempo, il SID trasforma la sua uscita di impulso in 0.



Rappresentazione del campionamento di un segnale. Il segnale continuo S(t) è rappresentato con una linea verde mentre i campioni discreti sono indicati con linee verticali blu.

Fonte: https://it.wikipedia.org/wiki/Campionamento_(teoria_dei_segnali)

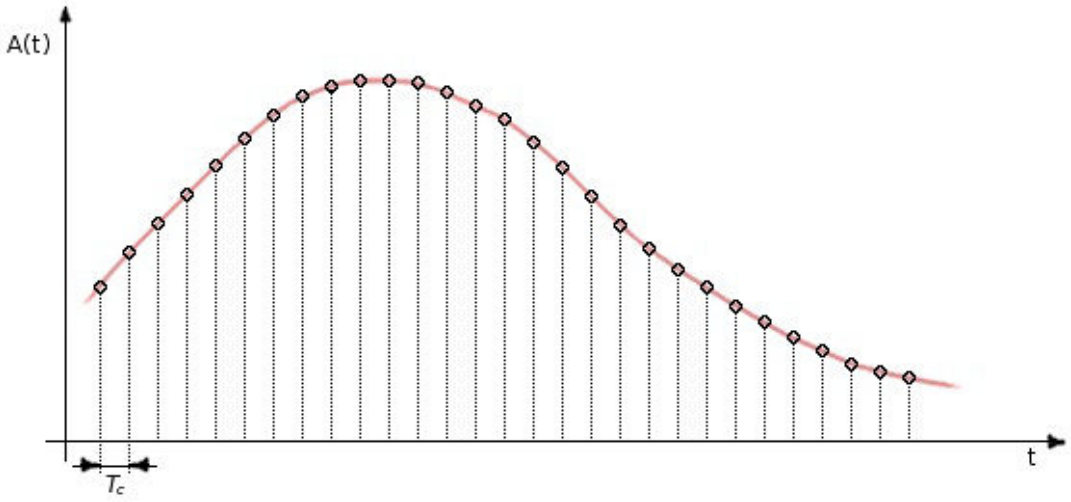
Per una spiegazione più approfondita, suggerisco una attenta lettura dei seguenti articoli:

o "Pulse Width Modulation, continued" by various.
Tying up some loose ends from last issue's digi article. ---> c=hacking issue 21

o "The C64 Digi", by Robin Harbron <macbeth@psw.ca>, Levente Harsfalvi <levente@terrasoft.hu>, and Stephen Judd <sjudd@ffd2.com> ---> c=hacking issue 20 + ADDENDUM con esempio

Questi articoli sono consultabili a questo indirizzo: http://www.ffd2.com/fridge/chacking
e contengono, tra l'altro, anche una spiegazione matematica del funzionamento del campionamento audio.

E' superfluo forse aggiungere, ma lo faccio lo stesso, sottolineare la bontà di questo lavoro, della mole di studio e test che avrà certamente comportato la sua realizzazione, tutto frutto di Levente H_rsfalvi e di altri, tutti già citati nei crediti contenuti nel codice.



Campionamento forma d'onda
Fonte: https://otolabdidattica.wordpress.com/2011/02/23/glossario-campionamento/

Ma cosa intendiamo per campionamento audio?

Il campionamento è una tecnica che consiste nel convertire un segnale continuo nel tempo oppure nello spazio in un segnale discreto, valutandone l'ampiezza a intervalli temporali o spaziali solitamente regolari. In questo modo, a seguito di una successiva operazione di quantizzazione e conversione, è possibile ottenere una stringa digitale (discreta nel tempo o nello spazio e nell'ampiezza) che approssimi quella continua originaria.

Se indichiamo con fc la frequenza di campionamento, ovvero la frequenza con la quale estraiamo i campioni ed indichiamo con fm la frequenza massima dello spettro del segnale da campionare, il teorema del campionamento di Shannon-Nyquist ci dice che se:

fc > 2*fm

allora la nostra frequenza di campionamento ci permetterà di ricostruire il segnale analogico che intendiamo riprodurre senza alcuna perdita di informazione.

Qualora la frequenza di campionamento che abbiamo adottato non dovesse rispettare tale condizione si riscontrerà un effetto conosciuto con il nome di "aliasing" che comporta una distorsione del segnale analogico ricostruito.

Riassumendo il concetto in parole più semplici: se preleviamo campioni alla giusta frequenza otterremo un segnale audio di qualità soddisfacente, altrimenti la qualità potrebbe non essere delle migliori.

Non entro in ulteriori dettagli per non annoiare troppo il lettore ma...ricordatevi di questo 'aliasing'.

Negli articoli che Vi ho citato troverete buona parte di ciò che possa servire a chi volesse addentrarsi ulteriormente. Inoltre, se le fonti già citate non dovessero rivelarsi sufficienti... Google è vostro amico, tenetelo sempre presente! :D

Come Vi accennavo già, ho anche aggiunto la gestione della EasyFlash, basandomi sul suo relativo SDK.
Tali info le ho reperite a questo indirizzo:
https://codebase64.org/doku.php?id=base:code_sample

La EasyFlash permette però di avere a disposizione fino ad 1 Mb di RAM, che permettono di riprodurre circa 1 minuto di brano.
Per superare tale limite ho adattato il codice del player per adoperare due cartucce di nuova generazione, la gmod3 e la gmod4, che consentono, almeno sulla carta, di poter avere a disposizione fino a 16 Mb di RAM!

Purtroppo sembra che la gmod3, che viene emulata da VICE nel momento in cui scrivo, non verrà sviluppata fisicamente.
Invece, in base anche alla conversazione via email con Jens Schönfeld, di Individual Computers, che sta realizzando questo nuovo hardware per il C64, pare che verrà portata avanti la gmod4, fino alla sua realizzazione fisica e quindi fino alla sua commercializzazione.

Riflettiamo ora sul campionamento che viene effettuato

ad una frequenza di 14400 Hz.
Secondo quando accennato (teorema del campionamento di Shannon-Nyquist) si riuscirebbero a ricostruire segnali con frequenza fino a 2*fm.
fm, ovvero la massima frequenza dello spettro del segnale da campionare varrebbe 7200 Hz (14400/2).
Cosa succede se nel brano audio da campionare sono presenti segnali a frequenza più alta?
Frequenze più alte non avrebbero abbastanza campioni a descriverle ma quegli stessi campioni descriverebbero una frequenza più bassa.
Questa frequenza viene detta frequenza di aliasing ed essendo bassa rientrerebbe nella banda dell'udibile e dunque avremo aggiunto al segnale iniziale una frequenza che prima dell'operazione di campionamento non esisteva mentre avremo perso quella frequenza alta che era presente nel segnale originario.

Oltre al problema dell'aliasing vi sono altri fattori che contribuiscono ad alterare la resa sonora del brano campionato. Uno di questi fattori è che non abbiamo infiniti livelli per descrivere ciascun campione ma solo 256 (ovvero adoperiamo solo 8 bit per effettuare il nostro campionamento).
Infine, c'è il problema del segnale "portante" che è sempre presente quando si modula un segnale,
Puoi spostare tale segnale verso frequenze non udibili oppure applicare un filtro passa-basso.

Anche l'autore del sorgente che Vi ho presentato si era posto il problema di eliminare i "fischi" che purtroppo qualcuno sente perché si trovano all'interno della banda delle frequenze audio che possono essere udibili dall'orecchio umano (20-20000 Hz circa)
Nell'articolo dell'autore viene suggerito di diminuire il livello del sustain sul SID per ridurre questi fischi fastidiosi ma ciò non è sempre vantaggioso e, inoltre, diminuisce il volume medio del segnale audio prodotto. Viene accennato anche all'impiego di un filtro passa-basso, di cui troviamo qualche traccia (commentata) anche nel codice ma, sebbene (con qualche modifica) funzioni, diminuisce drasticamente la qualità dell'audio riprodotto poiché esalta troppo le basse frequenze, dimezza le medie ed abbassa notevolmente le frequenze più alte. In effetti un buon filtro passa-basso fa esattamente questo :)
L'autore, nel suo articolo, si auspicava che con un procedimento simile a quello da lui descritto, si potesse arrivare a riprodurre un brano audio con una frequenza maggiore (magari con una risoluzione inferiore) per spostare tali fischi verso frequenze più alte e quindi non più udibili.
Credo sia questa la soluzione migliore, ma è stato necessario rivedere ancora una volta il codice, naturalmente! :D
Studiando il codice del player che ho riprodotto in questo articolo scoprirete che le voci che impiego sono 2 e non 3. Ciò mi permette di risparmiare istruzioni (e quindi tempo macchina) e pertanto riesco a salire verso frequenze di campionamento più alte.

Qualcuno di Voi starà già pensando...e se si utilizzasse una sola voce? Beh, l'ho provato e posso dirvi che...funziona! Arrivo a campionare a frequenze ancora più alte, risparmiando codice e quindi cicli macchina.
Il rovescio della medaglia è che ho sempre 1 Mb di RAM a disposizione con la EasyFlash e, con l'aumentare della frequenza di campionamento, crescono anche il numero di campioni necessari per la riproduzione di ogni secondo del brano e non riesco più a riprodurre 1 minuto di brano.

Approfitto di queste ultime righe per ringraziare il mio amico di vecchia data Davide Scuderi, che mi ha coadiuvato nei diversi test su hardware reale relativamente ai vari "spezzoni" audio che ho pubblicato, sia sul mio profilo personale Fb sia sul mio canale YouTube.

Se non fossi stato ragionevolmente certo della qualità della riproduzione di tali spezzoni su hardware reale, probabilmente questo mio lavoro sarebbe rimasto dentro un cassetto, perché sarebbe stato un mero esercizio di stile o poco più.

Prima di salutarvi, ho pensato di mostrarvi anche il codice sorgente del player così come l'ho rivisto e modificato nella sua versione che gira su gmod3. Tale versione non ha richiesto una modifica eccessiva del codice per renderlo funzionante sfruttando le caratteristiche di questa cartuccia. In questo caso specifico, ovvero sfruttando la gmod3, ho potuto mantenere il codice del player separato dal sample audio e lancio la riproduzione agganciando al player l'immagine virtuale della cartuccia che contiene quindi soltanto i campioni audio.

Tale approccio con la EasyFlash e la gmod4 non è stato possibile. Il codice del player risiede infatti nella cartuccia insieme ai campioni audio in entrambi i casi, rendendo il

codice decisamente più complesso.

A proposito, prima che lo dimentichi, ringrazio Jens Schönfeld, di Individual Computers ed il suo staff per il prezioso aiuto offerto nel migrare il codice del player dalla EasyFlash fino alla gmod4, con l'augurio che si riesca a commercializzare presto questa nuova cartuccia, riscuotendo tutto il successo che merita.

Di seguito il codice del player, modificato e rivisto opportunamente per riprodurre i campioni audio presenti su una cartuccia gmod3, scritto rispettando la sintassi Kick Assembler.

```
////////////////////////////////////////////////////////////////////////////

.label GMOD3_BANK0 = $de00
.label GMOD3_CTRL  = $de08

.const f_sample=$A0
.label f_sample_loc=$02
.label backup=$03
.label backup_x=$95

.pc=$0801

:BasicUpstart(main)

.pc=$080d

main:
alfa:
          jsr grafica

          lda #0
          sta $de00
          sta $de01
          sta $de02
          sta $de03
          sta $de04
          sta $de05
          sta $de06
          sta $de07
          sta $de08

          sta banklo+1
          sta bankhi+1

          lda max_contat_0
          sta contat_0

leggi_tasto:
          lda $DC01                 //READ SPACEBAR IF
          cmp #$EF                  //NOT PRESSED
          bne leggi_tasto     //THEN JUMP TO MAIN
                                    //PROMPT
          sei

          lda #$37        // Bank out kernal and basic
          sta $01         // $e000-$ffff

          lda #$7f        // Disable CIA IRQ's
       sta $dc0d
       sta $dd0d
       asl $d019      // Ack any previous raster interrupt
       bit $dc0d      // reading the interrupt control registers
       bit $dd0d      // clears them

ss:
          ////////////////////////////////////////////////////////////////
          //legge configurazione registri SID e la applica, registro per
registro

          ldx #$1C
ssid:
          lda sid_conf,x
          sta $d400,x
       dex
          bpl ssid

          lda $d011
          and #%11101111
       sta $d011                  //blank screen

       lda #$00                   //start address of sample in memory
       sta loop+1
       lda #$80
       sta loop+2

       lda #f_sample                    //end address high. Zpg ($fb) is
used
          sta f_sample_loc             //because of timing reasons

       ldx #$49                   //pre-load values for the SID command
       ldy #$41                   //registers...
          stx backup_x

omega:
          .fill $100-(omega-alfa)-$0d,$ea

loop_x:
          ldx backup_x
          lda #$80                      //start address of sample in
memory -HI
       sta loop+2

loop:  lda $face    //4            //Load next sample...
       sta backup
       sta $d402    //8            //put to the sample width reg LO
       stx $d404    //12           //RESET SID voice
       sty $d404    //16           //...then start wave counter

       //adc #$02     //18         //Increase sample value
          nop

       sta $d409    //22           //(presumably, there'll never be a
       stx $d40b    //26           //carry), then go on for next SID
       sty $d40b    //30           //channels.


       inc loop+1   //50           //Increase sample pointer
       bne loop1    //52                (when jumps, 53)

       inc loop+2                        //58       6
          lda loop+2                     //62       4
       cmp f_sample_loc                  //65       //cmp in pagina zero
ritarda 3 cicli.. 3
       bne loop                          //68    2 oppure 3 quando salta


          ////////////////////////////////////////////////////////////////

       inc banklo+1
       bne no_c

          inc bankhi+1
no_c:
          ////////////////////////////////////////////////////////////////

          ////////////////////////////////////////////////////////////////

          lda backup   //4              //Load next sample...

       sta $d409    //22           //(presumably, there'll never be a
       stx $d40b    //26           //carry), then go on for next SID
```

```
        sty $d40b    //30               //channels.

        //adc #$02     //18               //Increase sample value
           nop

           sta $d402    //8                //put to the sample width reg LO
        stx $d404    //12               //RESET SID voice
        sty $d404    //16               //...then start wave counter

           ////////////////////////////////////////////////////////////////

           ////////////////////////////////////////////////////////////////
bankhi: lda #$00                 // upper 3 bits of the bank number
        and #7
        tax

banklo: lda #$00                 // lower 8 bits of the bank number
        sta GMOD3_BANK0,x
           ////////////////////////////////////////////////////////////////


           lda backup            //sto leggendo un sample 0?
           bne cnt                          //se no, si continua a
ciclare....
           inc contat_0                     //se si incrementa contat_0
           bne cnt                          //se contat_0 non si è ancora
azzero continua a ciclare
                                            //altrimenti si
esce!
exit:
           lda #%00111011                   //bitmap mode on,Screen on,
normal screen contents are visible,screen height=25 rows
        sta $d011

        jmp main

cnt:
           jmp loop_x             //3

loop1:
        nop
        nop
           nop

           inc $d020   //6

           jmp loop

//-------------------------------------------------------------------------

grafica:
{
        ldx #0
lp2:
        lda screen+256*0,x
           sta $0400,x
        lda colors+256*0,x
           sta $d800,x

           lda screen+256*1,x
           sta $0500,x
           lda colors+256*1,x
           sta $d900,x

           lda screen+256*2,x
           sta $0600,x
           lda colors+256*2,x
           sta $da00,x

           lda screen+256*2+$e8,x
           sta $06E8,x
           lda colors+256*2+$e8,x
           sta $dae8,x

           dex
           bne lp2
           lda #<hires
           sta $72
           lda #>hires
           sta $73
           lda #$00
           sta $74
           lda #$20
           sta $75

           ldy #0
lp3:
           lda ($72),y
           sta ($74),y
        dey
           bne lp3

           inc $73
        inc $75

           lda $75
           cmp #$40
        bne lp3

           lda #0
           sta $d020

           lda #0
           sta $d021

           lda $d018
           ora #$08
           sta $d018

           lda $d011
           ora #$20
           sta $d011

           lda #$18   //multicolor mode on
           sta $d016  //
           rts
}

sid_conf:
.byte $0   //$D400
.byte $40  //$D401
.byte 0                 //$D402
.byte 0                 //$D403
.byte 0                 //$D404
.byte 0                 //$D405
.byte $f0  //$D406
.byte 0                 //$D407
.byte $40  //$D408
.byte 0                 //$D409
.byte 0                 //$D40a
.byte 0                 //$D40b
.byte 0                 //$D40c
.byte $f0  //$d40d
.byte 0    //$d40e
.byte 0    //$d40f
.byte 0                 //$D410
.byte 0                 //$D411
.byte 0                 //$D412
.byte 0                 //$D413
.byte 0                 //$D414
.byte 0                 //$D415
.byte 0                 //$D416
.byte 0                 //$D417
.byte $0f  //$D418
.byte 0                 //$D419
.byte 0                 //$D41a
.byte 0                 //$D41b
.byte 0                 //$D41c

contat_0:  .byte $FD
max_contat_0:          .byte $FD
```

```
.pc=$1000
screen:
*=* "screen"
          .import binary "res/titanic/screen.dat",2

colors:
*=* "colors"
       .import binary "res/titanic/colors.dat",2

.pc=$4000

hires:
*=* "hires"
          .import binary "res/titanic/hires.dat",2
```

//

In questo codice potrete notare che ho eliminato l'utilizzo di una voce, al fine di risparmiare cicli macchina e salire quindi in frequenza di campionamento.
Ho previsto la "ripetizione" della riproduzione dell' ultimo sample al cambio della pagina di RAM letta dalla cartuccia per passare alla successiva.
Se non venisse fatto, si sentirebbe un fastidioso "click" ogni secondo circa, in sottofondo.
Nel codice originale il dato letto veniva incrementato prima di passare alla voce del SID successiva.
Ciò però portava a distorsioni quando, passando dal valore numerico iniziale a quello finale, si passava da un valore $ff ad un valore $00. Quindi, ho commentato l'addizione, mantenendone però i cicli macchina necessari.

In giro potrete notare diverse ottimizzazioni per migliorare i tempi di esecuzione complessivi.
Con questa istruzione per esempio:
.fill $100-(omega-alfa)-$0d,$ea
faccio girare il codice del player all'interno della stessa pagina di RAM di 256 bytes.

A questo link potrete trovare un video che fa vedere in azione l'algoritmo che ho adoperato per i campionamenti mediante modulazione PWM(acronimo di Pulse Wave Modulation), in BASIC V2 (moooolto lento, naturalmente):
https://youtu.be/f5MpDSbvYwM
Penso sia davvero tutto ora. Spero di non averVi fatto annoiare troppo.
Se qualcuno dei lettori desiderasse ulteriori delucidazioni e/o informazioni scriva pure una mail alla redazione di RetroMagazine (retromagazine.redazione@gmail.com) e proverò, nei limiti del possibile naturalmente, ad accontentarlo.
Un caro saluto a tutti Voi, amici e lettori di RetroMagazine.
A presto!

Riproduzione musica campionata su Commodore 64 - "Total Eclipse of the Heart", B.Tyler (1983) https://www.youtube.com/watch?v=zx6Fv8hIb0o

Riproduzione musica campionata su Commodore 64 - "Rasputin", M. Boney (1978) https://www.youtube.com/watch?v=-4NMAdRKIq4

Riproduzione musica campionata su Commodore 64 - "Wonderwall", Oasis (1995) https://www.youtube.com/watch?v=ZJPRqEOUDcc

Il peso del Tetris

Intervista ad Alexey Pajitnov, il creatore di un gioco leggendario

di Ksenia Kamash (XENIA KAMA) – Trad. Carlo Nithaiah Del Mar Pirazzini

La streamer e youtuber russa **Xenia Kama** (che ci aiuterà in redazione) è riuscita ad intervistare il celebre sviluppatore.

XENIA KAMA: Alexey, buongiorno. Probabilmente lei è uno degli sviluppatori di giochi più noti dello spazio post-sovietico in generale. "Tetris" è stato definito il gioco più popolare al mondo. Sente il peso della sua popolarità perché ha creato un gioco così famoso?

Alexey Pajitnov: Sì, certo, lo sento. Ma ci ho fatto l'abitudine perché il gioco esiste da 40 anni. È diventato rapidamente molto popolare, quindi potresti dire che ho avuto successo molti anni fa. E poi è diventata una specie di routine. Ho rilasciato molte interviste e presentato vari premi per le competizioni "Tetris". Ho ricevuto un bel po' di posta dai fan appassionati di "Tetris" che chiedevano autografi. Mi inviano tutti i tipi di cartucce, foto e carte da firmare. Questo è il mio fardello. Ma mi piace.

XK: "Tetris" ha una sua storia. Ce ne vuole parlare?

AP: "Tetris" ha avuto origine da un puzzle chiamato "Pentomino". Sembra una scatola quadrata di 5×12 o 6×10, contenente tutti i possibili pentamini. Un pentamino è una forma composta da cinque quadrati. Possono essere ordinatamente incastrati insieme in vari modi, formando lettere come Z, X e così via. Quando ho iniziato a creare giochi simili sul computer, ho avuto un'idea: creare un gioco per due giocatori basato su questo set.

XK: Cosa faceva al Computing Center durante la creazione di "Tetris"?

AP: Lavoravo nel laboratorio di cibernetica tecnica al Computing Center dell'Accademia delle Scienze dell'Unione Sovietica. Concentrandomi principalmente su microprocessori e sistemi a microprocessore. Più precisamente, nel settore del riconoscimento automatico del parlato. Stavamo sviluppando un dispositivo di estrazione delle caratteristiche del parlato che era collegato al computer BESM-6: potevamo immettere frammenti del

Ad Alexey piace davvero il suo gioco... lo porta proprio "con se"

Alexey in una foto di repertorio

parlato nella memoria BESM-6 e registrarli su nastri, quindi riconoscerli utilizzando vari metodi. Ora, probabilmente suona divertente, ma a quel tempo sembrava fantascienza.

XK: Ci racconti come "Tetris" è arrivato sulla scena mondiale.

AP: Il primo editore del gioco è stato un uomo britannico di origine ungherese di nome Robert Stein. Lo vide all'istituto SZKI, come raccontò in seguito. Era una specie di Computing Center che si occupava di problemi informatici a Budapest. Stein andava lì e vedeva che software avevano, ne acquistava alcuni, ne pubblicava altri. Vide "Tetris" lì e gli piacque. Stipulammo un contratto formale con Stein. Probabilmente lo ha interpretato come un pieno accordo sulla licenza e ha iniziato a pubblicare "Tetris".

A quel tempo, né le persone fisiche né le persone giuridiche potevano avere contatti con organizzazioni straniere. Solo agenzie specificamente organizzate si occupavano di tali contratti. Il Ministero del Commercio Estero aveva diverse agenzie che acquistavano e vendevano vari articoli all'estero. Le ho visitate tutte durante quei pochi mesi in cui mi occupavo di questo. Sono stato respinto da Licensintorg, non sono mai arrivato a RFTA e altrove mi hanno chiesto di non disturbarli. Nessuno è riuscito a parlarmi, perché erano tutti troppo occupati. Ma ho incontrato un ragazzo piuttosto simpatico all'ELORG, che si è interessato con entusiasmo al mio "Tetris". Mi sono sentito sollevato perché la questione era nelle mani di qualcuno dell'ELORG e ho iniziato a scrivere nuovi giochi e a concentrarmi su altre cose.

Dopo un po', si è scoperto che "Tetris" era già stato pubblicato e ho ricevuto rimproveri per aver inviato un telex errato. Ho detto loro che avevo risposto nel miglior modo possibile e che avrebbero potuto verificarlo di persona. Mi hanno spiegato che ciò che avevo scritto era corretto, ma che non era giusto che una persona inesperta comunicasse, poiché la mia risposta è stata interpretata come un via libera alla pubblicazione, anche se non l'avevano ancora ricevuto.

Alla fine, tutto questo pasticcio commerciale era già pronto e alla fine dell'88, questa versione di "Tetris" fu rilasciata

su tutti i PC in America e in Europa.

Nell'88, "Tetris" ottenne i primi posti in tutte le competizioni di videogiochi. Il gioco fu distribuito su decine di computer domestici: Sinclair, Commodore 64, Amiga, Atari, ecc. Mirrorsoft lo pubblicò ovunque. Spectrum HoloByte realizzò un'ottima versione per PC. E poiché il gioco, come sapete, crea una forte dipendenza, divenne rapidamente popolare. Ma non abbiamo concesso i diritti per la console. Il contratto iniziale copriva solo i computer, dispositivi con un monitor e una tastiera. Le console avevano una TV e un joystick, quindi non erano incluse nell'elenco.

A quel tempo, il processo di gestione dei diritti non era ben definito, quindi Stein vendette questa versione a qualcun altro. Alla fine fu pubblicata e quando il mio amico Hank Rogers venne qui per i diritti del Game Boy e portò la cartuccia "Tetris" pubblicata per Nintendo, gli dissero: "Ehi, non abbiamo dato a nessuno questi diritti". In realtà, fu una fortuna che si trovasse lì perché avevamo appena risolto quel problema. Si dimostrò piuttosto solido, serio e onesto negli affari, il che ci permise di assicurarci un ottimo contratto per le macchine portatili, le piattaforme a cui apparteneva il Game Boy. Ecco perché l'uscita del Game Boy divenne un evento significativo nella storia dei giochi per computer.

Emigrazione e affari negli USA

XL: Come si è trasferito negli USA. Oggigiorno, storie come la tua sono piuttosto comuni nel mondo IT, ma all'epoca immagino fosse piuttosto insolito.

AP: Ad un certo punto, ho capito che lavorare sul riconoscimento vocale era un vicolo cieco, una scelta sbagliata fin dall'inizio. Ciò che mi ha davvero attratto è stato l'eccellente accesso che avevo al computer BESM-6. Ho cercato di trovare un'attività che in qualche modo si allineasse ai miei interessi di gioco e che avesse un aspetto legittimo.

Fu allora che mi balenò l'idea che i giochi non sono prodotti esclusivamente tecnici e che, se hanno un legame con un campo della scienza, è molto probabile che sia la psicologia. Sono andato dagli psicologi, al dipartimento di psicologia, da conoscenti di mio padre che erano anche psicologi, e

Hank Rogers e Alexey Pajitnov

El Fish di Maxis in tutto il suo splendore

ho detto: "Ragazzi, sono un bravo programmatore, ho accesso al computer. Forse posso fare qualcosa per voi? Ci sono delle metodologie o qualcosa che vi interessa? Come possiamo sviluppare la psicologia con l'aiuto dei computer?"

Tutti erano interessati a questo. Ho finito per avere due o tre contatti commerciali e alla fine abbiamo computerizzato le loro metodologie. Ci ho messo un sacco di impegno e ho guadagnato popolarità nella comunità di psicologia informatica. È così che ho incontrato il mio futuro socio in affari, Vladimir Pokhilko. Si è scoperto che entrambi siamo stati invitati a creare giochi e abbiamo collaborato con successo un paio di volte. Ci è piaciuto lavorare insieme e lui è diventato il mio partner di gioco più o meno permanente.

Vladimir mi convinse con insistenza a trasferirmi. Lasciammo l'URSS nel gennaio del '91. Durante la nostra assenza, il paese cessò di esistere, non avemmo nemmeno la possibilità di guardarci indietro. Tecnicamente, non avevamo un posto dove tornare, ma il paese che ci aveva rilasciato i passaporti e ci aveva mandato a lavorare non c'era più. Non pensammo di tornare perché c'erano un sacco di opportunità, dopotutto, era per questo che eravamo venuti. Più tardi, ricevemmo le nostre green card e ci sistemammo.

XK: Quindi, ha avviato la sua attività con Vladimir Pokhilko e si è dedicato allo sviluppo di giochi negli Stati Uniti.

AP: Non abbiamo solo sviluppato giochi, ma anche tecnologie per giochi. Il nostro primo progetto non era esattamente un gioco; si chiamava "El Fish", un acquario elettronico con pesci. Maxis lo pubblicò su PC e in seguito su altri dispositivi. Creare un acquario visivamente gradevole sullo schermo era un compito impegnativo a quel tempo. I pesci si riproducevano persino e si potevano incrociare. Era un progetto interessante.

XK: Si considerava un imprenditore? Le piaceva farlo?

AP: A dire il vero, non proprio. È stato gratificante realizzare di poterlo fare, ma non mi è piaciuto particolarmente. Provavo un senso di orgoglio quando riuscivo a creare qualcosa di nuovo al computer. Tuttavia, quando riuscivo a concludere affari commerciali, non provavo molta soddisfazione. Ecco perché non mi sono mai considerato un uomo d'affari serio. Certo, potevo organizzare qualcosa se necessario. Dopotutto, ho trascorso un anno e mezzo

come presidente di un'azienda, e in quel periodo non è andata in bancarotta. Capisco cosa sono gli affari e come funzionano. Ma onestamente, non mi piace farlo e non voglio farlo.

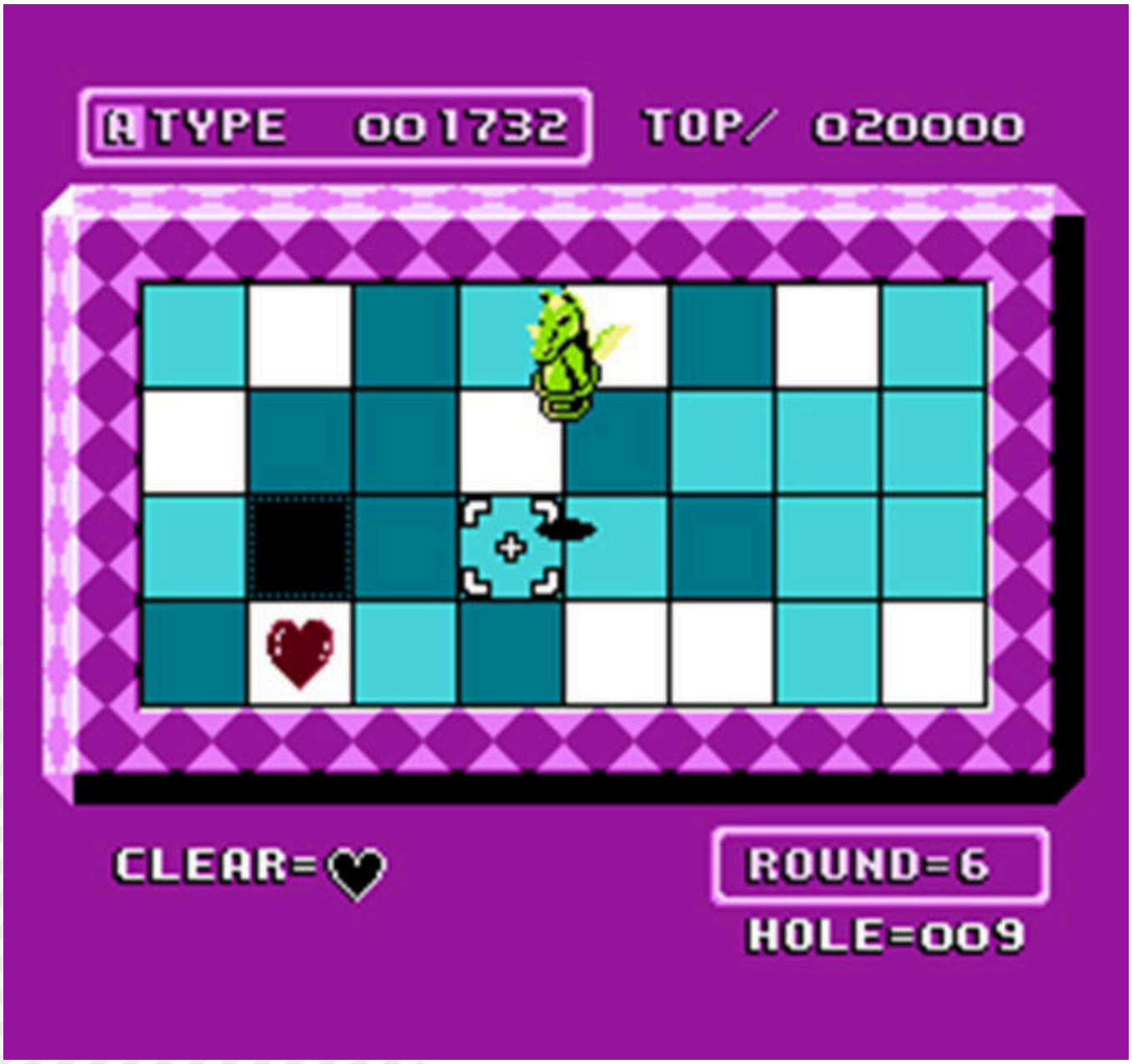
Il numero di giochi sviluppati da Alexey Pajitnov

XK: Ho provato a contare quanti giochi ha sviluppato, ma non ci sono riuscita. In parte perché ce ne sono parecchi, e in parte perché ci sono molte varianti di "Tetris". Non è chiaro se siano giochi separati o parte di una famiglia. Quindi, quanti giochi ha sviluppato?

AP: Non so come affrontare neanche questo. È ancora più difficile per me contarli perché li amo tutti. Sia quelli che sono stati ampiamente pubblicati sia quelli che hanno avuto uscite limitate o sono stati creati per caso. Ad esempio, ho sviluppato io il gioco "Knight's Tour". È un gioco con pezzi degli scacchi, veri, ed è uno dei miei preferiti. La joint venture "Dialog" me l'ha comprato nell'87. Mi hanno dato un computer con un disco rigido da 80 megabyte: ero felice perché finalmente avevo il mio computer. Poi questo gioco è stato pubblicato una volta, non ci crederai, su un disco per Nintendo.

Per un po', il Famicom ha avuto un sistema su piccola scala su un disco. Il disco aveva una traccia a spirale. Era quasi come un nastro magnetico ma su un disco.

Il gioco rilasciato si chiamava "Knight Move", che era il mio "Knight's Tour". Poi, con la mia approvazione, è stato



Knignt Move/Knight's Tour in versione Nintendo Famicom Disk System

pubblicato e ripubblicato da Spectrum HoloByte. Si sono troppo immersi nella grafica 3D e nei dettagli eccessivi, il che ha fatto sì che il gioco rallentasse. Di conseguenza, non ha venduto affatto. Quindi, uno dei miei giochi preferiti non è stato pubblicato correttamente.

In realtà, è uno dei primi giochi resta tra i miei preferiti. L'ho realizzato poco dopo "Tetris", intorno all'86 o all'87. A quel tempo, ho creato circa cinque giochi. La maggior parte di essi erano inclusi in "Microsoft Entertainment Pack: The Puzzle Collection" per MS-DOS. Ne sono molto orgoglioso. È un po' antiquato per gli standard odierni, con un'interfaccia un po' macchinosa, ma è una buona raccolta di 10 giochi di puzzle onesti e molto originali. Sei di questi erano miei e gli altri quattro sono stati creati da qualcun altro. In quella collezione c'è un gioco chiamato "Muddled Casino". È stato il mio primo gioco in assoluto: un puzzle game.

L'ho inventato prima ancora di "Tetris" per l'Electronika 60. Poi i ragazzi l'hanno portato su PC, migliorandolo notevolmente pur mantenendo l'idea originale. Anche se è un po' pesante, provo dei sentimenti positivi nei suoi confronti perché è stata la prima cosa che ho creato.

I migliori giochi

XK: Qual è il miglior gioco per computer per lei, indipendentemente dal genere?

AP: Adoro i puzzle e ogni due mesi circa, navigo nell'App Store alla ricerca di qualcosa che mi catturi. Anche se, onestamente, navigarci è deludente perché le offerte sono piuttosto primitive e non ce ne sono molte interessanti. C'era "Monument Valley", un bellissimo gioco di puzzle, in bianco e nero. E anche un gioco abbastanza decente chiamato "Empires and Puzzles".

Lo gioco ogni mattina a piccole dosi da sette anni e ho già raggiunto i massimi livelli possibili. Tuttavia, non ho sentimenti particolarmente calorosi nei suoi confronti; è più un'abitudine.

Un altro è "The Witness", una raccolta di puzzle combinati con un'esplorazione 3D. Cammini attraverso uno spazio meticolosamente progettato e incontri schermate di puzzle che devi risolvere. I puzzle vanno da molto semplici a estremamente impegnativi. Poi mi piace Warcraft. Amo quella saga.

XK: Se non le dispiace, che livello ha raggiunto in "Warcraft"?

AP: Non ero tra i primi 100 giocatori, ovviamente, ma probabilmente ho ottenuto tutto ciò che una persona normale potrebbe ottenere in "Warcraft". Mi ci sono appassionato e poi ho fatto amicizia con persone che giocavano insieme. Ma "Warcraft" mi interessava anche come designer, quindi a differenza dei giocatori normali, ho adottato un approccio più ampio. Ho studiato giocatori e personaggi e ho cercato di esplorare tutte le missioni per ogni specializzazione nel gioco. Ho giocato non solo per vincere, ma anche per non perdermi nulla.

Un consiglio da un ragazzo normale che vuole approcciarsi al mondo dello sviluppo software

XK: Ci stiamo gradualmente avvicinando alla fine. Spero che la nostra intervista venga letta dai futuri sviluppatori di giochi. Ha qualche consiglio per loro?

AP: Sono solo un “ragazzo” normale, non un genio, un guru o un da Vinci, qualcuno che ha inciampato molte volte. Ho sperimentato molti fallimenti e ho creato giochi sia brutti che belli, di successo e non. Odio dare consigli, soprattutto in un campo creativo come lo sviluppo di giochi. Quindi posso dare solo un consiglio: ascolta te stesso.

Le cose migliori e più importanti possono essere realizzate quando non si cerca di fare qualcosa di meglio degli altri, ma, al contrario, si mira a creare qualcosa da cui non si può staccarsi. Questo approccio forma uno sviluppatore serio.

Giappone 25^puntata: Gachagacha...pon!

di Michele Ugolini

Cari lettori siete pronti per una puntata elettrizzante sui Gachapon?
In questa seconda puntata dedicata ai gachapon parlerò del contenuto "adulto" di alcune macchinette in Giappone. Si, avete capito bene, tra le tante meravigliose miniature che si possono collezionare, i giapponesi hanno brillantemente pensato che potrebbero essere collezionabili anche oggetti relativi al mondo delle luci rosse. Chiaramente attraverso una modalità diversa dalla nostra interpretazione occidentale, quindi... procediamo con ordine.
Come già parlato nel precedente articolo, in Giappone da alcuni anni è ritornata la moda dei Gachapon. Noi collezionisti italiani ricorderemo gli Exogini: miniature di Wrestler provenienti dallo spazio, una moda nata dall'originale idea giapponese dei Kinkeshi.
Oltre alle miniature di questi pupazzetti in Giappone esistono intere linee di produzione di miniature riguardo pressoché qualsiasi oggetto presente nella nostra fantasia: miniature di utensili da cucina, miniature di animali, miniature di mascotte, etc..
Quindi... quali miniature mancano a questo infinito elenco? Mancano le miniature delle parti anatomiche umane nonchè gli annessi giocattoli per adulti da associare alle varie parti anatomiche, che la nostra fantasia e perversione potrebbero dar vita nella nostra testa. Secondo voi ho acquistato una palletta di questi gachapon per adulti? No, non l'ho fatto, ma al prossimo viaggio... lo farò!
Questi gachapon si trovano lungo le maggiori strade turistiche dello shopping, esattamente come le macchinette classiche fruibili dal pubblico. L'unica differenza dai gachapon classici è la posizione di queste macchinette per adulti: sempre a fianco di un negozio aperto. Questi negozi li ho unicamente visti in funzione durante il giorno, quindi non sono sicuro che durante l'orario notturno di chiusura sia permesso lasciare queste macchinette incustodite fuori dal negozio. I gachapon per adulti sono fruibili unicamente da maggiorenni e la particolarità grafica della macchinetta è che non vi sono immagini del contenuto delle varie pallette. C'è una bella scritta rossa su sfondo nero e la palletta che uscirà conterrà un oggetto per adulti, alla modica cifra di 500 yen.
Questi gachapon per adulti li ho visti nel quartiere turistico di Namba poco distante da Osaka. Apparentemente e, soprattutto, personalmente, in base alle mie tappe del viaggio, non ricordo di averne trovati a Tokyo.
Per dovere di cronaca, nonchè dei blog, annovero la presenza di queste macchinette per adulti, chiamate anche "Eropon", installate sul pianerottolo a fianco le scale del piano per adulti di Nobunaga Shoten a Teramachi. Oltre alla grafica particolare degli Eropon, anche la capsula della palletta è particolare: è completamente nera e non puoi sapere cosa c'è dentro finché non la tiri fuori dal serbatoio delle pallette. Sulla macchinetta c'era scritto "Per soli 500 yen puoi ottenere il sorriso più erotico della tua vita. Non temere nulla, prendilo!" Dai blog leggo che alcuni premi sono piuttosto costosi e quando ricevi un buono puoi scambiarlo con un premio alla cassa. Negli Eropon che ho visto c'erano molte capsule nere vuote appoggiate in un cesto lateralmente alla macchinetta, quindi l'acquisto di queste pallette sembra essere piuttosto popolare. Oppure, forse, i proprietari di queste macchinette hanno messo le pallette aperte appositamente per mostrare ai futuri acquirenti il successo di quel prodotto.

Vi allego un link del Nobunaga Shoten così potrete curiosare riguardo le fantasie erotiche nipponiche. https://www.e-nobunaga.com/
Inoltre vi farò vedere una carrellata di immagini di tutte le "sorprese" dentro le pallette oscurate, ovviamente in modalità nipponica, estremamente perversa ed altamente pixellata. Sto scherzando! Ripensandoci.. una immagine di una mutandina sexy possiamo concederla alla nostra rivista. Non c'è niente di male rispetto a ciò che in realtà potrebbe esssere contenuto in altre pallette nere. Ed ecco a voi finalmente la risposta dopo questi infiniti dubbi ed una effervescente fantasia che è già andata in orbita. Una mutanda sexy! Si, in queste pallette, di questo preciso Eropon c'erano delle mutande, femminili, probabilmente carine, sicuramente scomode, composte da tessuto sintetico, poco "green". Chissà a quanti lavaggi riusciranno a resistere? Chissà se saranno mai indossate? Chissà quante persone che hanno inserito le monete, vorrebbero rientrare in possesso delle monete stesse? Ecco intanto due link che potete consultare riguardo la mappa in città di questi gachapon.
https://islaywhiskey.hatenablog.com/entry/2018/02/10/103133
https://logblog.blog33.fc2.com/blog-entry-650.html
Probabilmente questa tipologia di distributori è presente nelle maggiori città, soprattutto nelle vie dedicate ai turisti, vie piene di cianfrusaglie che possiamo trovare in qualsiasi negozio di articoli cinesi sotto casa. Amici, nel frattempo che facciamo atterrare la fantasia, dopo un breve volo, vi invito a visitare il Giappone ed a scoprire l'anima del Sol Levante, partendo dalle loro origini, le loro magiche tradizioni millenarie fino a, perché no, scoprire cosa contengono queste oscure pallette. Alla prossima lettura cari amici, si intitolerà Gachagachagacha-pon, parleremo delle particolarità di queste misteriose e frequentatissime macchinette.
A presto!

Fireteam Rogue per Super Nintendo, l'ambizione senza un'azione non è nulla.
Il racconto di un titolo dal potenziale incredibile!

di Takahiro Yoshioka e Evan Granovitz - Carlo Nithaiah Del Mar Pirazzini

Fireteam rogue era un promettente gioco action-adventure sviluppato tra il 1994 e il 1999 da Accolade che purtroppo non vide mai luce.

Una delle cose più impressionanti per questo gioco è la quantità incredibile di concept art disponibile, che dimostra la qualità e la passione messa in questo prodotto.

Quasi tutti i disegni del gioco furono realizzati dalla compianta artista visuale Betty Cunningham e sono disponibili sul suo sito web alla pagina: http://www.flyinggoat.com/main.html.

Fig. 1 - Il personaggio di Shadowblade dell'artista Betty Cunningham

Ambizione eccessiva

La cancellazione di Fireteam Rogue è dovuta all'eccessiva ambizione. Gli sviluppatori (o meglio, le persone che si occupavano del marketing del gioco) erano allo sbando con affermazioni roboanti: 100 ore di gioco, personaggi più grandi della maggior parte degli altri action game, voci digitalizzate, parlato in gioco e molto altro ancora.

Tutto questo, unito alla pessima gestione del team, ai continui litigi interni e alla fine dei fondi portarono alla catastrofe.

Russell Borogove (uno degli sviluppatori) afferma tutt'oggi: "*il progetto ha avuto problemi in fase di sviluppo. Abbiamo trascorso tutto il tempo a sviluppare schermi con dati ridicoli per adattare i livelli di gioco, quando in realtà avremmo dovuto tagliare nemici e livelli per riuscire a renderlo concreto.*

Ci sono stati anche conflitti personali che sono culminati con l'abbandono del produttore del progetto proprio quando eravamo all'uscita della versione beta. Poco dopo, il responsabile del prodotto presso Accolade si chiese se il gioco valesse tutto questo sforzo e quanto tempo ancora occorresse per terminarlo. Non mi era chiaro quanto lavoro ci sarebbe voluto per finirlo e se il titolo fosse in grado di competere sul mercato e quindi decidemmo di chiudere tutto".

Fig. 2 - Uno dei livelli completi del gioco disponibile con la versione Alpha

La già citata Betty Cunningham sul suo sito web affermava che il gioco fosse completo.

Il graphic designer Scott Ruggels ricorda: "Entrambi i progetti di gioco (Fireteam Rogue e il gioco Megadrive mai pubblicato, Cybernauts: The Next Breed) erano diretti da John Skeel. Non so cosa gli sia successo dopo Accolade, ma dopo che il budget di 2 milioni di dollari per il gioco fu speso, con circa 750.000 spesi in materiali promozionali, tra cui un rotolo di plastica alto 6 piedi con un'immagine

a grandezza naturale del personaggio principale generata al computer all'interno, e il prototipo, che, in tutta onestà, non era molto divertente da giocare, il gioco fu cancellato, insieme a molti altri subito dopo che la nuova gestione prese il sopravvento, e prendemmo strade separate, con me che finii alla società 3DO durante gli ultimi giorni di sviluppo di "Real Player".

Il titolo fu definitivamente abbandonato nel 1996 ma il progetto andò avanti un paio d'anni.

Marketing

Fireteam Rogue era un gioco sontuoso e pomposo. Durante i primi anni '90, ci fu un nuovo boom del mercato dei fumetti (in parte dovuto alla Image Comics e alla "rinascita" di DC e Marvel).

Accolade intendeva capitalizzare anche su questo settore. Creò un fumetto che introduceva la storia. Un fumetto di quindici pagine che sarebbe uscito assieme al manuale del gioco stesso su SNES.

Oltre al fumetto, c'erano il progetto delle action figure e persino un programma televisivo (su questo punto non è chiaro se live action o animato).

Soldi su soldi su soldi. A questo aggiungiamo anche un ologramma realizzato per un tour chiamo EGM Super Tour che si concentrava presso convention di fumetti e grandi centri commerciali.

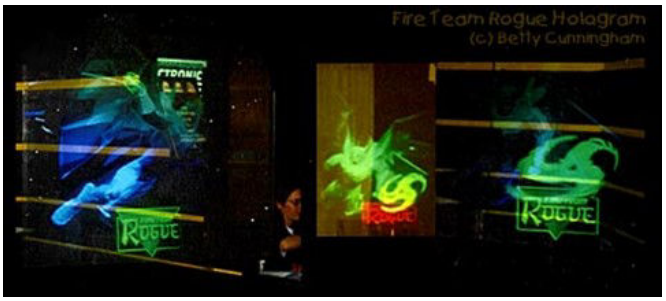


Fig. 3 - Ologramma utilizzato all'EGM Super Tour

Sui media del tempo Fireteam Rogue fu accolto in modo "tiepido". La rivista Nintendo Power lo mostrò nel febbraio 1994. In una lunga anteprima mostrano qualche immagine del gioco e notano la "limitata tavolozza di colori che ricorda i titoli per Megadrive".

Game Players nell'ottobre 1994 elogia il titolo con "serio potenziale" e che i personaggi di quelle dimensioni erano innovativi.

La stessa Nintendo Power, ma nel 1995 critica la lentezza di sviluppo e la strana grafica a scatti del gioco.

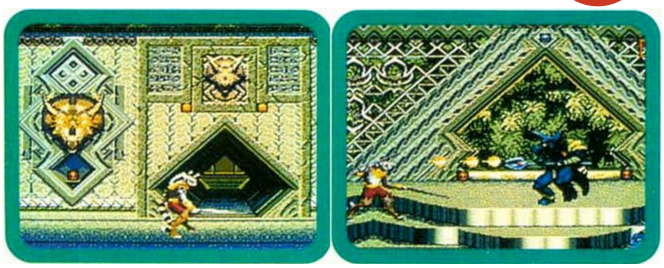


Fig. 4 - Prime immagini di Fireteam Rogue dal numero di Nintendo Power di febbraio 1994 (scansionate da Retromags)

Il primo prototipo

Il prototipo si rivela alla comunità retrò nel 2005 e mostra l'incompletezza dello sviluppo alla fine del 1994.

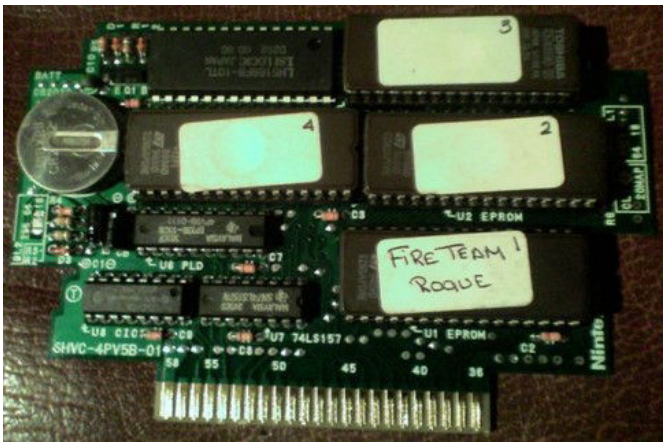


Fig. 5 - Il primo prototipo della cartuccia

L'immagine rom è disponibile su diversi siti del settore tra i quali SNES Central.

Mostra diversi livelli ma solo uno sembra completato correttamente, quello di Shadowblade.

Gli effetti sonori sono piuttosto minimalisti, anche se la

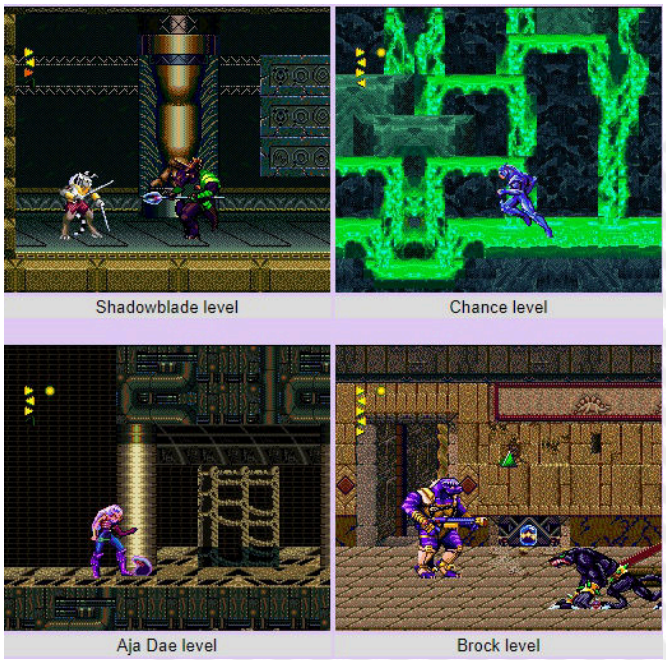


Fig. 6 - Un esempio dei livelli disponibili nel prototipo

musica è ben fatta. Il gioco è minimalista: nessuna vita extra o continue. Animazioni un po' a scatti e i controlli sono appena decenti. E' presente persino un lag nello spostamento del personaggio.
Grafica insipida e una tavolozza limitata dei colori.

Prototipo Successivo

Il secondo Alpha arriva nel 2010 e mostra un livello di sviluppo del tardo 1995.
Quasta Alpha 2 mostra uno stato di completamento più avanzato ma lo stesso gameplay della precedente.
Il livello di Shadowblade viene leggermente modificato rispetto alla prima versione mentre gli altri tre livelli mostrano stati di completamento aggiuntivi ma restano sempre incompiuti.
L'aggiunta principale di questa versione è l'uso del MODE 7 per alcune fasi di gioco.
Nella fase di volo si deve saltare dentro ad un velivolo. Piuttosto riuscita.
Queste fase di shooting sono una gradevole novità in un immenso ammasso di ambizione.
Sono state aggiunte le password per poter prosegure nel gioco ma mancano ancora i continue.

Fig. 7 - Qui possiamo vedere la fase di volo

La fine

Come abbiamo visto tutto terminò tra liti e mancanza di fondi. Fireteam Rogue avrebbe tutte le carte in regola per essere un titolo promettente.
Una bella trama fantasy, una grafica (in apparenza) impressionante per il tempo e una spinta di marketing altissima.
I livelli sparatutto funzionavano abbastanza bene e si potevano confrontare senza problemi con altri titoli analoghi per lo Snes.
Detto tutto questo le ROM Alpha disponibili mostrano però un gioco che non possiamo definire "giocabile".
Un prodotto con un tempo di sviluppo enorme e decisamente fuori contesto. Il Super Nintendo era ancora attivo sulla scena ma erano già da tempo sul mercato le console di nuova generazione, e il popolo voleva vedere quello.
L'ultima Alpha fu mostra al CES invernale del 1995 e fu massacrata dalla critica.
Level design scadente, grafica scattosa e poche idee. Tutto questo era Fireteam Rogue
La lezione che ci può insegnare il gioco è che concentrarsi troppo sull'Hype e sulla storia prima della creazione di un solido gameplay può affondare la nave.
La storia rivaleggiava con la maggior parte degli RPG del tempo e le idee potevano indirizzare il titolo verso una versione "moderna" di Super Metroid.
Invece, un ciclo di sviluppo impantanato da una cattiva gestione, ritardi e roboanti proclami rese questo titolo un'altra nota a piè di pagina nella storia dei 16bit.

Le rom le potete trovare qui:

Prima beta:
https://mega.nz/file/17YA2ZSb#YaDOt0d8rHIuH7NOdsACY8a6GXieHz9m5fsmwa_Yyk8
Seconda beta:
https://mega.nz/file/sqBDWZDb#MO742MjE9ZWReEwv3585kjG6MU1Smk4Hu0ncHLXyA8E

Fonti:

- Sito web di Betty Cunningham con molti concept art.
- Intervista con Jim Barnett, in cui si parla di vari giochi Accolade (tra cui Fireteam Rogue) al Consumer Electronics Show dell'estate 1994 a Chicago.
- Electronic Gaming Monthly, Anteprima - include diversi screenshot, Data di pubblicazione: settembre 1994, Volume: 62, Pagina: 150.
- Gamepro, Anteprima (include quattro screenshot, nessuna delle scene è presente nei due prototipi), Data

di pubblicazione: febbraio 1995, Volume: 77, Pagina: 157. - Nintendo Power, Pak Watch (anteprima, menziona l'ambizioso gameplay e la trama e la "limitata tavolozza di colori". Mostra due screenshot del livello Shadowblade e concept art di Broc.), Data di pubblicazione: febbraio 1994, Volume: 57, Pagina: 111

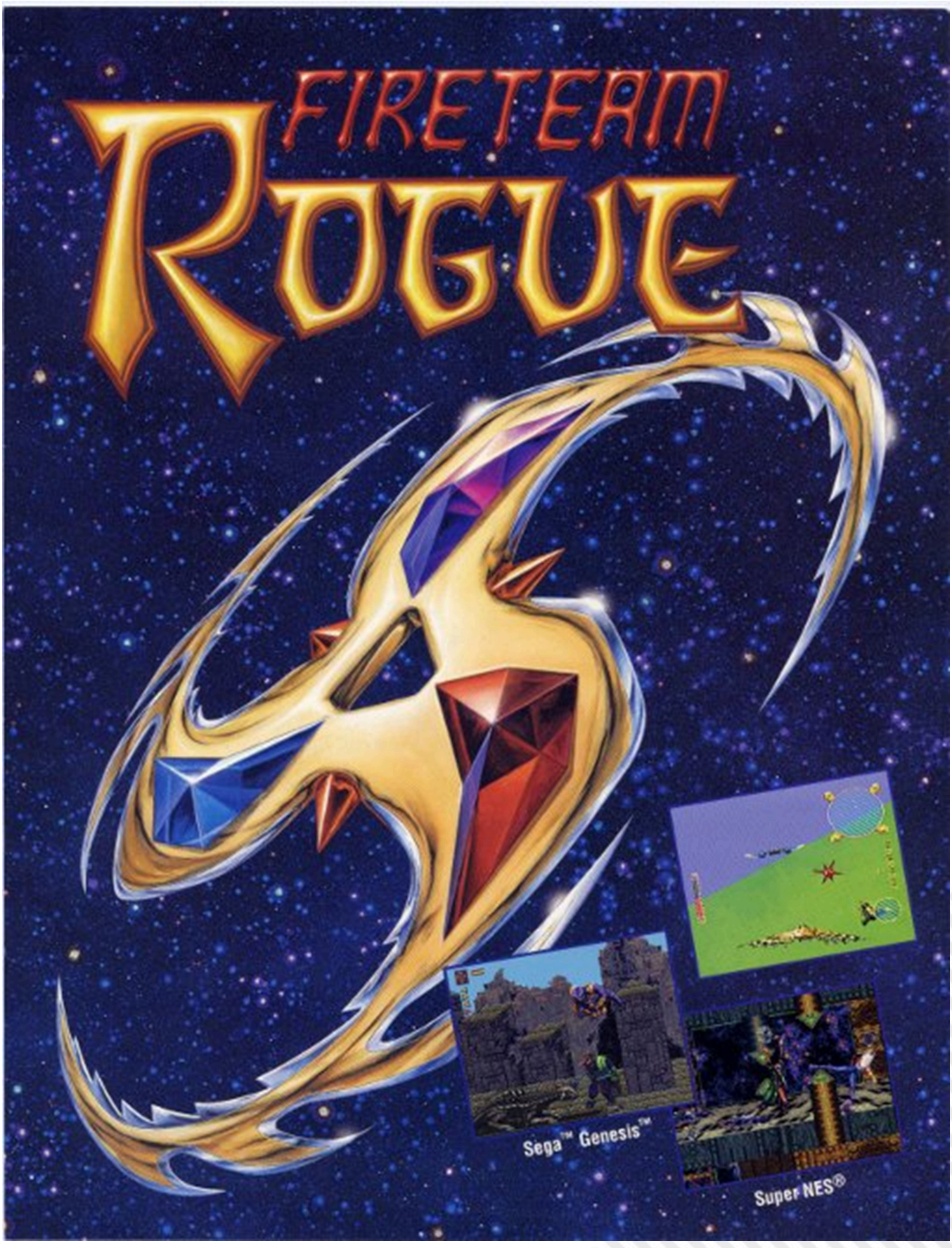


Fig. 8 - Il foglio di presentazione del gioco al WINTER CES del 1995

FIRENZE VINTAGE BIT 2024

a cura di Associazione Firenze Vintage Bit Onlus (scritto da Alberto Apostolo)

Per tutti gli appassionati e tutti i collezionisti di retrocomputer, siamo lieti di annunciare che **Domenica 1° Dicembre 2024 a Lastra a Signa (Firenze)** si svolgerà la **sedicesima edizione del Firenze Vintage Bit**.

La manifestazione sarà tenuta quest'anno nei locali assegnati all'interno della Scuola Primaria Santa Maria a Castagnolo situata in Via Gramsci 139 nel Comune di Lastra a Signa (Firenze).

L'intento è quello di far divertire gli espositori e il pubblico di tutte le età mostrando modelli di computer e console che hanno fatto la Storia dell' Informatica e macchine calcolatrici che di solito si vedono nei musei.

L'ingresso è **libero e gratuito** (valgono le norme della buona educazione e del buon senso nell'accedere ai locali e alle postazioni degli espositori).

Si ricorda che **il Firenze Vintage Bit non è un mercatino** e che non vi sono fini di lucro. Appassionati e collezionisti desiderosi di effettuare qualche scambio, dovranno recarsi fuori dall'area riservata all'evento.

Come si svolge

Si inizia il Sabato che precede l'apertura al pubblico. Gli espositori, regolarmente iscritti, portano i loro computer e console, preparano i tavoli a loro assegnati, collegano le periferiche verificando le connessioni. E' una occasione per provare anche le macchine di altri espositori, scambiare opinioni e suggerimenti sul software o su eventuali adeguamenti dell'hardware.

La giornata si conclude con una cena conviviale presso un locale caratteristico della zona.

La giornata di Domenica (dalle ore 9:00) è invece dedicata al pubblico di tutte le età che potrà provare le macchine esposte e fare domande agli espositori che saranno contenti di rispondere e raccontare divertenti aneddoti. La chiusura dalle 13:00 alle 14:30 è riservata alla pausa per il pranzo insieme (prenotazione obbligatoria comunicata prima della manifestazione allo staff organizzativo).

L'evento termina alle ore 18:00 per facilitare il rientro degli espositori arrivati da più lontano.

Il Firenze Vintage Bit non si limita alla sola esposizione. Ogni edizione è associata a un tema, ovvero un argomento conduttore da illustrare.

Il tema di questa edizione è "Simulatori e Didattica".

Al mattino è prevista una conferenza intitolata "Un percorso nel mondo della simulazione" di 45 minuti circa, nel quale si offre una panoramica delle diverse tipologie di simulazioni con applicazioni al contesto didattico.

Nel pomeriggio vi è lo spazio per un altro breve intervento (sul gioco di strategia al computer "The Sumerian Game") seguito da un saggio musicale (colonne sonore di video giochi).

Breve storia delle passate edizioni

Dopo avere partecipato nel 2007 come espositore al Varese Retrocomputer, Walter Pugi matura l'idea di creare una manifestazione nel Centro Italia (possibilmente a Firenze). Con il prezioso aiuto di Maurizio Morandi, nel 2009 a Campi Bisenzio (Firenze) si è tenuta la prima edizione del Firenze Vintage Bit ottenendo un inaspettato successo (replicato ancora nel 2010 e nel 2011). Dal 2012 al 2022 (tranne l'edizione 2020 organizzata come video call causa COVID-19), le edizioni si sono svolte nel suggestivo ambiente medievale dell'Antico Spedale di Sant'Antonio di Lastra a Signa (Firenze). Nel 2023 è stata ospitata

Fig. 1: FVB 2009. Gruppo fondatori FVB con alcuni ospiti. Da sinistra: Stefano Donati e Luca Bisti che portarono il prototipo dell'emulatore RealSpectrum da loro realizzato, Walter Pugi (ideatore e fondatore), Gianni Zamperini (promotore), Maurizio Morandi (fondatore) e Alberto Sarcinelli (fondatore).

Fig. 2: FVB 2010. Tavolo di un espositore.

Fig. 3: Antico Spedale di Sant'Antonio.

a Signa (Firenze) nei saloni della magnifica Villa Alberti.
Invece, quest'anno è ritornata di nuovo a Lastra a Signa, avendo ricevuto la disponibilità della Scuola Primaria Santa Maria a Castagnolo (SMAC).
Come detto in precedenza, anche le passate edizioni sono state caratterizzate da un tema, affrontando argomenti come grafica, musica, programmazione e gli anniversari dei modelli più iconici. Infine, non sono mai mancate le esibizioni musicali e corali molto applaudite dagli ascoltatori.

L'Associazione Firenze Vintage Bit

Costituita nel 2016, ha ricevuto la qualifica di ONLUS (Organizzazione Non Lucrativa di Utilità Sociale) nel 2017. L'associazione supporta l'organizzazione dell'evento ed è un interlocutore presso Amministrazioni Locali e altre istituzioni.
Con il passare degli anni, l'opera instancabile dei suoi Volontari ha arricchito il programma associativo, con diverse iniziative praticate nella sede di Lastra a Signa (corsi di Storia dell'Informatica, corsi di Programmazione, laboratorio di riparazione di computer e periferiche). Inoltre, l'Associazione collabora con le scuole ed è ospite in diversi eventi sociali a Firenze e in Toscana, ricevendo sempre un entusiastico gradimento da parte del pubblico.
Dal 2024, l'Associazione Firenze Vintage Bit Onlus è gemellata con l'associazione Varese Retrocomputer, in ricordo della prima esposizione fatta da Walter Pugi e per condividere idee e opinioni al fine di migliorare sempre di più l'offerta di attività ed eventi.
L'Associazione Firenze Vintage Bit Onlus ha un **gruppo su Facebook**
https://www.facebook.com/groups/firenzevintagebit
e un **canale su YouTube**
https://www.youtube.com/channel/UCUQzraVIBquit-KnFIRrDTw

Informazioni utili

Le informazioni sulla manifestazione (comprese indicazioni stradali e modalità di iscrizione per gli espositori) si trovano sul sito
www.retrocomputer.org

Fig. 4,5: FVB 2018. All'interno dell'Antico Spedale di Sant'Antonio.

Fig. 6: FVB 2018. Foto di gruppo degli espositori.

Fig. 7: Laboratorio riparazioni (Marzo 2023).

Fig. 8: Evento "Renai in Fiore" (Aprile 2023).

Fig. 9: Evento "Festa delle Scuole" presso la Scuola Primaria SMAC (Maggio 2024).

Con il patrocinio del
COMUNE DI LASTRA A SIGNA
Assessorato alla Cultura

Firenze Vintage Bit 2024

16° edizione

SIMULATORI E DIDATTICA

Graphic Design: Walter Pugi

Lastra a Signa
Scuola Primaria
Santa Maria a Castagnolo
1 dicembre (09:00-13:00 / 14:30-18:00)

Lastra a Signa
Città di Enrico Caruso

info@retrocomputer.org
www.retrocomputer.org

GRAFIKART FORNITURE PER UFFICIO
Radio Incontro
SPONSOR UFFICIALE
vignolbags

NEW GAME

QUINTY

Anno: 1999/2024
Editore/Sviluppatore: Game Freak
Genere: Puzzle game
Piattaforma: Super Nintendo

Il ramake inedito SNES del gioco di debutto di Game Freak (i creatori di Pokemon) è stato finalmente recuperato.

Quinty (noto anche come Mendel Palace negli USA) è stato originariamente rilasciato per Nintendo NES nel lontano 1989 ad opera dei creatori dei Pokemon e successivamente riportato su SNES con un upgrade tecnico alla portata della piattaforma a 16bit Nintendo.
Si tratta di un puzzle-action in cui il giocatore controlla un personaggio che capovolge delle piastrelle sul pavimento per sconfiggere i nemici. Uno strano modo di "combattere" che però risulta piacevole e dalle meccaniche interessanti.

La versione SNES di Quinty non fu mai rilasciata ufficialmente, rendendo questa riscoperta online particolarmente interessante per fan e per chi preserva il software per questa console.

Il recupero "illecito" di ottobre 2024 a danni dei Game Freak ha decisamente portato alla luce un buon titolo.
Esteticamente è ben fatto e colorato. Molto "kawaii" nell'aspetto e ben animato. C'è un buon sonoro ed è tutto piuttosto intuitivo anche se totalmente in lingua giapponese.
Il concetto di gioco è interessante e la meccanica, una volta appresa, coinvolge anche il più scettico (tipo me) dei giocatori.

Attenzione però! Il titolo è adatto agli appassionati di puzzle game e agli amanti del mondo nipponico nonché coloro che conoscono la versione NES/ Famicom (decisamente di nicchia in Europa).
Per tutti gli altri è un titolo da conoscere e preservare.

di **Roberto Del Mar Pirazzini**

GIUDIZIO FINALE

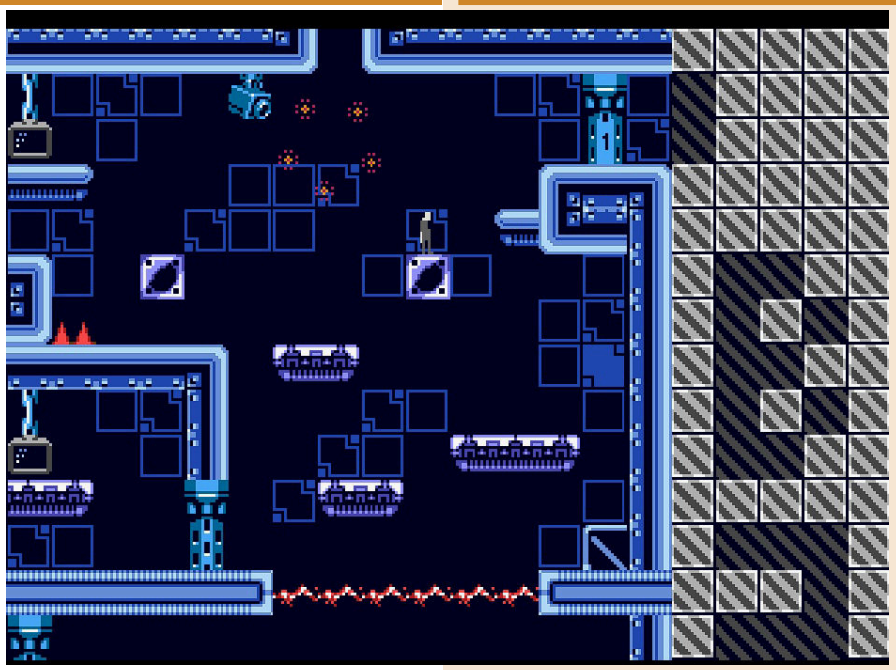
» Giocabilità 80%
La meccanica di gioco è molto simpatica e appassiona. Interessante la versione multigiocatore fino a 4 attraverso multitap.

» Longevità 75%
Il giapponese non è limitante ma sicuramente non è fruibile da subito. Alcuni livelli possono diventare monotoni.

NEW GAME

ESCAPE 2042

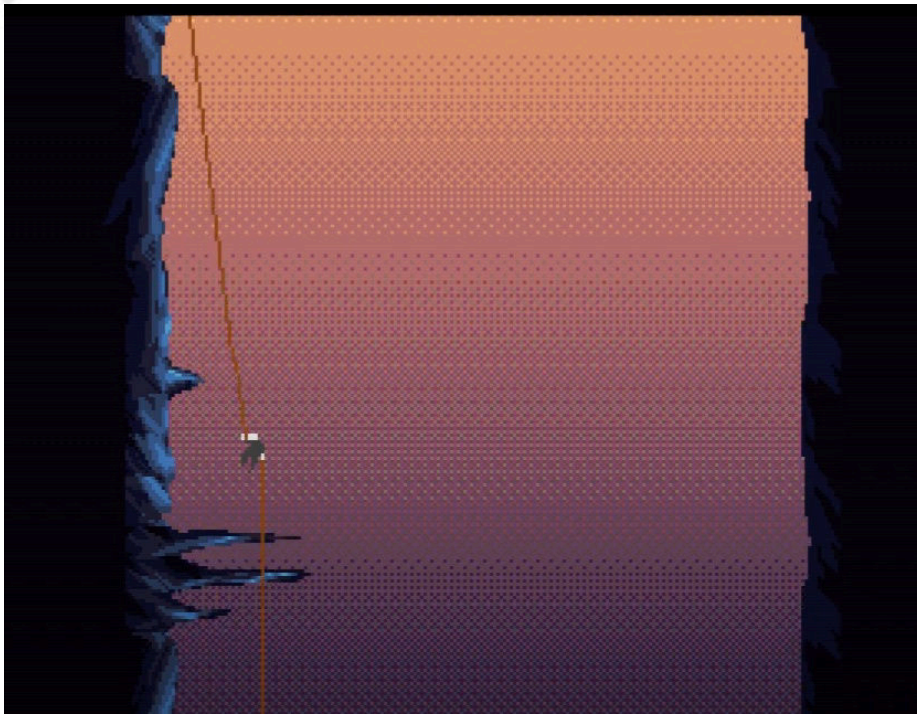
Anno: 2017
Editore/Sviluppatore: Orion Soft
Genere: Action/Puzzle
Piattaforma: Atari Jaguar
Sito web: https://www.orionsoft.games/



Nel 2042, un governo totalitario è al potere. L'elite governativa si nasconde dietro una democrazia fittizia, disconnessa dalla vita che conduce la gente e dedita al solo interesse.
Il regime considera ogni dissidente una cospirazione da eliminare. Chiunque non la pensi come loro viene considerato un terrorista e rinchiuso in prigioni di massima sicurezza.
Shun, ingegnere informatico e membro della coalizione Truth Defenders, rinchiuso nella prigione di Bulor 24 fugge.
Ovviamente, amici miei, saremo noi a guidare Shun in questa roccambolesca fuga.

Escape 2024 si svolge in tre ambientazioni differenti (prigione, deserto, foresta) ed è intervallato da due mini giochi, tra cui uno "shoot em up" e un action game puro. Lo scopo del gioco è sopravvivere evitando ogni contatto con i carcerieri.
Da evitare le telecamere di sicurezza, le guardie e le trappole. Lungo il percorso possiamo raccogliere alcuni oggetti utili come le granate oppure esercitare le nostre abilità di hacker bloccando il sistema informatico o disattivando i sistemi di sicurezza.
Sono inclusi due livelli di difficoltà. Normal che consente una via più "semplificata" alla risoluzione dell'avventura e Hard, dove non sono presenti armi per difenderci e le sequenze di minigioco durano il doppio. Un titolo che pare un platform ma invece racchiude in sé numerosi elementi e stili differenti.

Ho trovato la stessa tensione di Impossible Mission nei primi livelli di gioco ed è una cosa positiva.
Questa versione Jaguar è identica alle altre versioni uscite anche per Sega Megadrive, Dreamcast e altre piattaforme retro.



GIUDIZIO FINALE

» Giocabilità 90%
Un mix di stili differenti di gioco con diversi mini giochi e un bel grado di sfida.

» Longevità 85%
Titolo che vi terrà stretti alla vostra postazione di gioco per diverso tempo.

I controlli sono precisi e lo sviluppo di enigmi e livelli è ben strutturato ed intelligente. E' un titolo che non punisce mai severamente il giocatore ma richiede attenzione e studio del livello.

La grafica è molto raffinata. Una pixel art 8bit con accenti a 16bit. Difficile da spiegare, ma personalmente ho gradito molto questo stile artistico che si adatta perfettamente a Escape 2024.

Lodevole anche la colonna sonora che, pur non presentando troppo tracce, è davvero buona e pulita nel suono nitido e chiaro. Uno stile molto retro anni 80 che ho amato particolarmente.

Se nel 2017 vi siete persi questo piccolo gioiellino, rimediate subito. Escape 2042 merita di essere giocato sui vostri Jaguar.

di **Giampaolo Moraschi**

NEW GAME

KUDZU

Anno: 2024
Editore/Sviluppatore: MegaCat Studio
Genere: Action RPG
Piattaforma: Game Boy

Kudzu, l'ultimo Zelda-like di MegaCat Studio va oltre la semplice evocazione nostalgica di base. In questo titolo c'è tanto game design "da Game Boy". Sembra di giocare ad uno dei titoli del periodo glorioso della piccola console portatile.

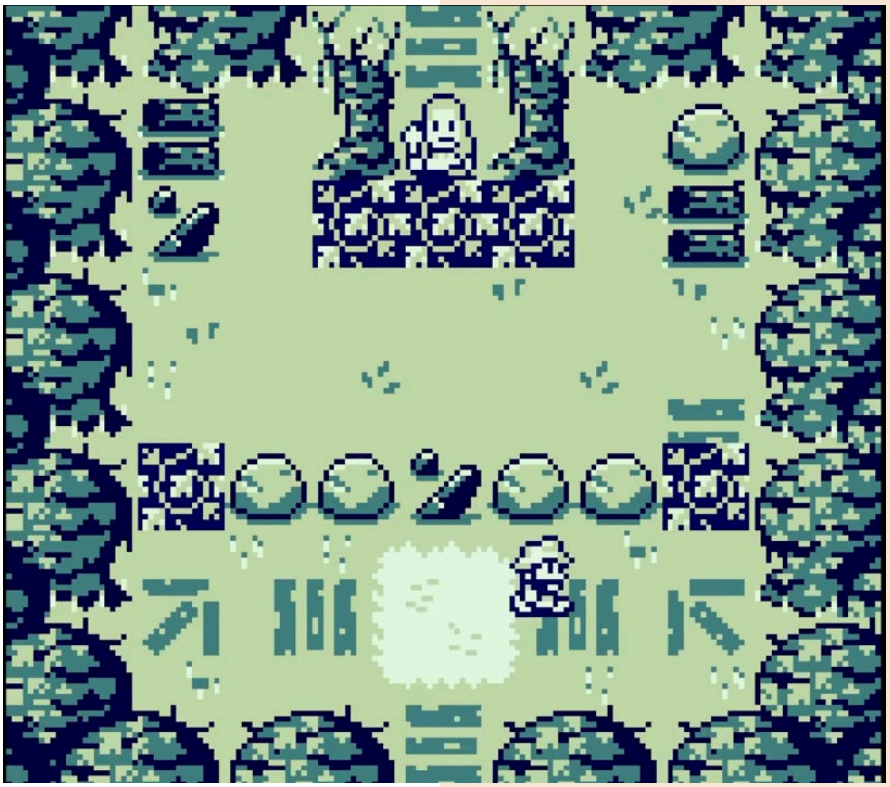
I giocatori che cercano qualcosa che vada oltre le vibrazioni generali dei primi giochi degli anni '90 vedranno il fascino di Kudzu, con la sua visuale dall'alto e le meccaniche facili da capire.

Nel titolo il giocatore veste i panni dell'apprendista giardiniere MAX, che intraprende un'avventura per salvare il suo mentore, Zoen, dalle pericolose piante di Kudzu che stanno infestando il pianeta terra.

Meccanicamente si gioca un po' come Link's Awakening. Man mano che il giocatore procede nella storia, acquisisce strumenti da giardinaggio che gli permetteranno di attaccare i nemici e di eliminare i pericoli ambientali.

Ogni nuovo strumento sostituisce il precedente e garantisce l'accesso a nuove parti della mappa, in modo molto simile ai mondi Zeldiani, offrendo al giocare nuovi percorsi, Le aree Dungeon richiedono una risoluzione degli enigmi e i combattimenti con i boss. Se i primi non sono difficili, devo ammettere che sconfiggere i boss di fine dungeon invece faranno sudare le proverbiali 7 camicie.

Ho trovato l'approccio narrativo di



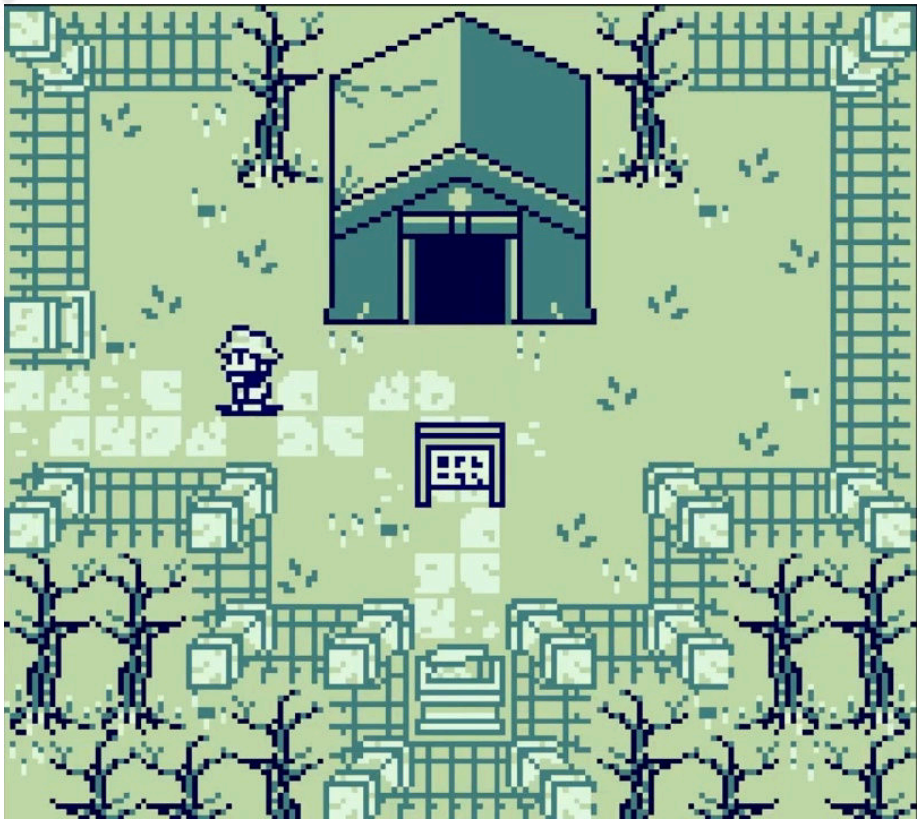
Kudzu e la progressione in gioco molto interessante. Si inizia il gioco, si scopre la quest di base e poi si parte all'avventura. La storia parte nel più semplice dei modi e diventa chiara via via che si gioca.

Il titolo risente purtroppo di una certa ripetitività negli scontri basici e questo frena l'entusiasmo dell'avventura e delle scoperte.

Tecnicamente è un prodotto ben sviluppato con una grafica varia e delle buone animazioni.

Sono presenti alcuni glitch in un paio di boss fight, ma nulla che va a compromettere la comprensione dello scontro.

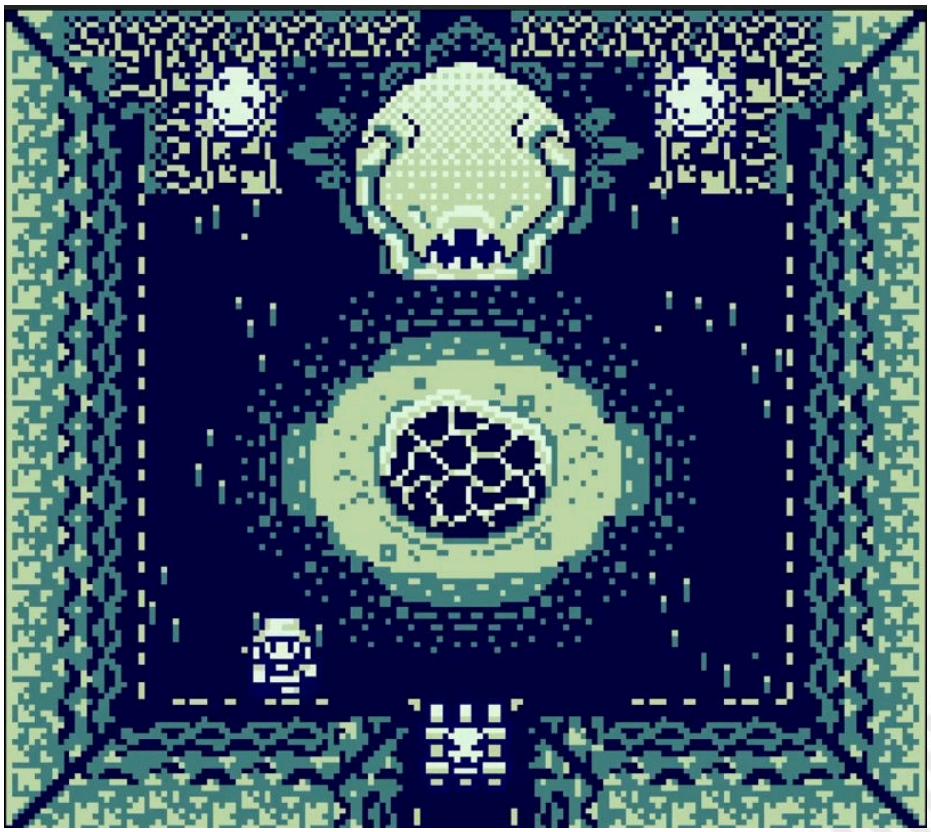
Simpatica la musica ma effetti sonori molto bassi e sotto tono, sembrano quasi assenti.



È un titolo con molto carattere e tantissimi dettagli e soprese e vi saprà guidare in una bella avventura.
Non sarà forse l'erede di Zelda, ma non pretende di esserlo.
Provatelo.

Sito web: https://megacatstudios.com/products/kudzu?srsltid=AfmBOopYqRDZu3ZUOUJgezFKPuDvq_jCVTCY4ZU2jZCpNCmVLCMXw2au

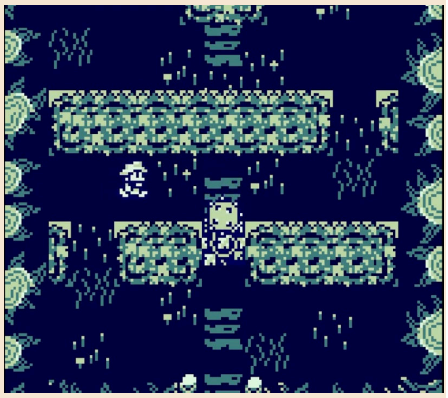
di **Roberto Del Mar Pirazzini**

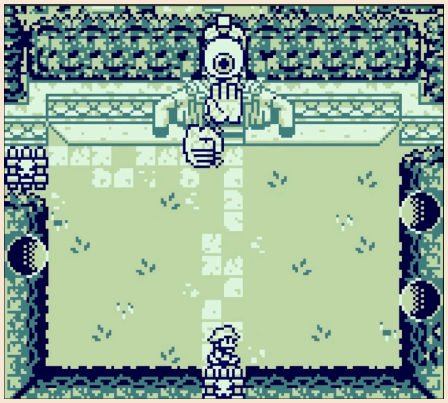


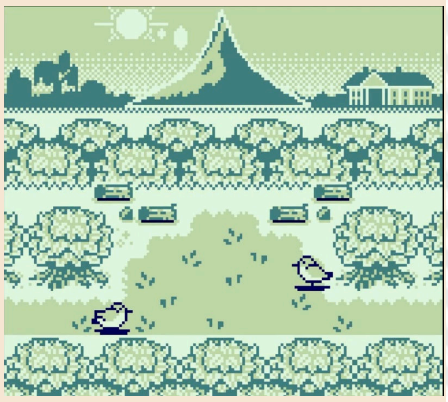
GIUDIZIO FINALE

» Giocabilità 75%
Una bella storia, scritta con cura e ben sviluppata in game.

» Longevità 70%
Le boss fight sono davvero toste (non tutte) e sono presenti alcuni glitch grafici non proprio gradevoli.
Manca anche uno stile "consono" degli effetti sonori.





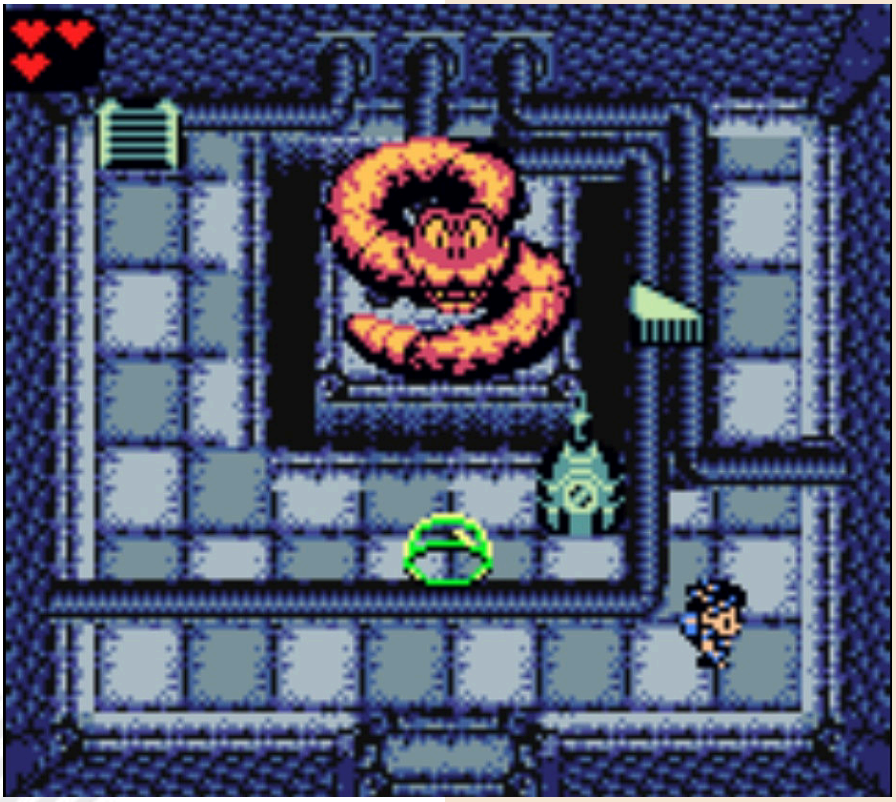


NEW GAME

ZEPHYR'S PASS

Anno: 2024
Editore/Sviluppatore: Incube8 Games, CRTOGRPHR
Genere: ACTION RPG
Piattaforma: Game Boy Color
Sito web: https://incube8games.com/products/zephyrs-pass-gbc-digital-edition

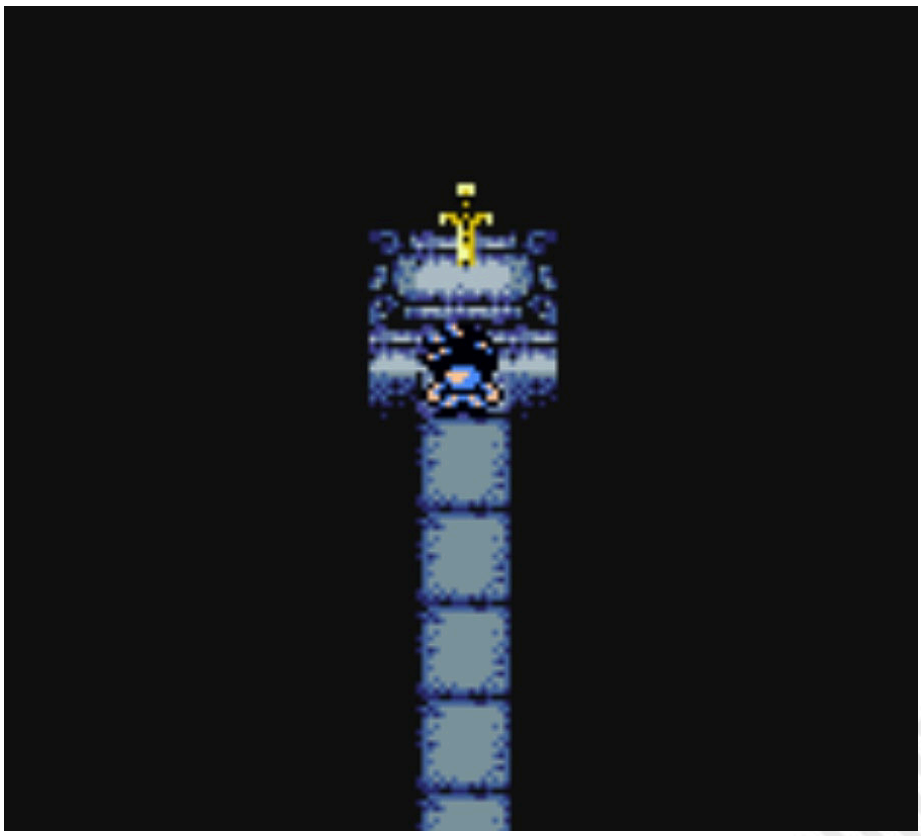
Zephyr's Pass è uno dei nuovi JRPG per console che mi ha più coinvolto ultimamente. Un titolo avvincente e con una narrativa bellissima, nato per essere giocato su Game Boy Color.
Mescola Final Fantasy, Zelda, eroici guerrieri, temibili e buffi pirati dei cieli, rapimenti, riscatti personali... Insomma un gioco con tantissima azione e che si lascia giocare con molta facilità.
Si parte nei vestiti di Abel, giovane eroe fattorino che si vede, suo malgrado, coinvolto in una storia molto più grande di lui.
Dovrà unirsi a dei pirati dei cieli per sconfiggere Thane, un folle despota che vuole sostituirsi al Dio del Vento. Per poterlo abbattere, Abel dovrà recuperare gli antichi pezzi dell'armatura del vento, ognuno dei quali conferisce nuove abilità e la forza per salvare la sua terra (e la bella di turno).
Disarmante nella sua semplicità di gioco e nell'uso dei pochi tasti a nostra disposizione, Zephyr's Pass si è rivelato una bellissima sorpresa tra i nuovi titoli in uscita in questi mesi.
Possiede un bel sistema di menù che ci permette, premendo SELECT, di tenere sotto controllo mappa, equipaggiamento, salvataggi e quest. Il tasto direzionale muove Abel, il tasto B usa la spada e il tasto A serve per difendersi ed utilizzare oggetti oppure, nel caso siano presenti personaggi non giocanti, dialogare.
Lo sviluppatore ha messo tutto il suo amore per la piattaforma Nintendo e per i grandi titoli del passato,



regalandoci un'estetica bellissima con l'uso dei pochissimi colori del Game Boy Color.
Le animazioni, le schermate narrative, la grafica dei fondali e dei personaggi sono davvero belle.
Ma il punto di forza del gioco è nella narrazione che ci prende e non ci lascia più. La storia di Zephyr's Pass vuole essere giocata tutta d'un fiato e vissuta. Gli enigmi in gioco sono ben studiati e i momenti d'azione semplici ma soddisfacenti.

Un titolo quasi esente da difetti disponibile in digital download e in formato fisico su cartuccia.
Ecco, forse in questo caso il vero difetto è il costo elevatissimo del supporto fisico. Comunque sia viaggiare tra le nuvole con la nave volante è un'esperienza unica. Senza indugio da provare.

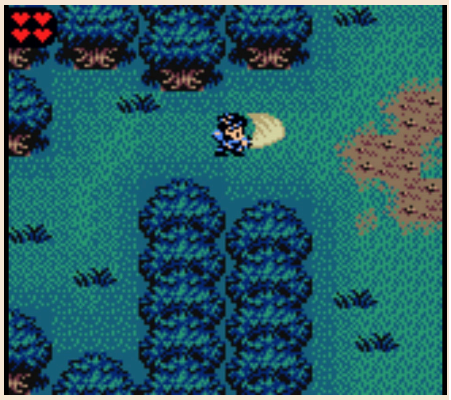
di **Carlo Nithaiah Del Mar Pirazzini**



GIUDIZIO FINALE

» Giocabilità 90%
Semplice, intuitivo e con buone meccaniche di gioco.

» Longevità 90%
La storia è ben scritta e difficilmente ci staccheremo dal gioco prima del termine.



NEW GAME

KING'S QUEST VI
HEIR TODAY, GONE TOMORROW

Anno:1992/2024
Editore/Sviluppatore: Marko/Atarimania
Genere: avventura grafica
Piattaforma: Atari ST
Sito web: https://www.atarimania.com/game-atari-st-king-s-quest-vi-heir-today-gone-

Bello rivedere King's Quest dopo tutti questi anni. La saga di Sierra On-line che impazzava tra fine anni 80 ed inizio anni 90.

Questo sesto episodio chiamato "King's Quest VI: Heir Today, Gone Tomorrow" usci per la prima volta nel 1992, scritto da Roberta Williams e Jane Jensen.

Un titolo riconosciuto come punto culminante della serie per il suo storico filmato 3D realizzato da Kronos Digital Entertainment e il doppiaggio con la voce dell'attore Robby Benson che forniva la voce al protagonista del gioco.

È stato anche l'ultimo titolo di Sierra Online ad essere pubblicato su floppy disk.

Sono quindi rimasto stupito nel rivederlo dopo tanti anni in versione

Atari ST. Una versione realizzata da Marko/Atarimania che ha preso i file originali della versione Pc e il launcer AGI Quest.

Si tratta di un demake dell'originale. Ci sono voluti 18 anni di lavoro per rilasciarlo (dal 2006 ad oggi) e il

GIUDIZIO FINALE

» Giocabilità 80%
Una delle saghe più iconiche per questo genere di giochi. Forse un po' datata nel gameplay ma sempre bellissima.

» Longevità 90%
C'è tantissimo da fare, esplorare e vedere. Compresi i finali multipli.

risultato è ottimo.

King's Quest Vi è stato assemblato e compilato utilizzando Agi Studio. Uno strumento di sviluppo utilizzato da Sierra On Line per i suoi primi giochi d'avventura.

Quando AGI debutto nei primi anni 80 era una tecnologia innovativa per quei tempi e fu rilasciato non solo su PC ma anche su Apple II e Apple IIGS, Amiga, Atari ST e Mac. Un motore di gioco semplice che comprendeva un linguaggio chiamato Game Adaptation Language, un compilatore e un interprete. Il linguaggio era ad alto livello e assomigliava al C. Questo veniva compilato in bytecode ed eseguito dall'interprete. Nel 1988, con il lancio del SIERRA'S Creative Interpreter, fu lentamente abbandonato.

Tornando al titolo possiamo dire che c'è tutto (o quasi). Il regno delle Isole verdi, il complotto, le quest e anche i finali multipli.

Sono presenti alcuni bug ma nulla di terribile, anzi lo sviluppatore invita chiunque a segnalare gli errori per poter migliorare con le prossime release.

La saga di King's Quest è una pietra miliare per le avventure grafiche e merita di essere giocata e riscoperta soprattutto in questa versione.

di **Giampaolo Moraschi**

SPELUNKY

Anno: 2004
Editore/Sviluppatore: PaulKo64
Genere: Platform
Piattaforma: Commodore 64
Sito web: https://paulko64.itch.io/spelunky64

Il platform indie acclamato dalla critica Spelunky è arrivato su Commodore 64. Lo sviluppatore principale del progetto è Paul Koller che ha lavorato al titolo assieme a Mikkel Hastrup che si è occupato della parte audio. Il gioco ha cominciato il suo sviluppo nel 2017 e lentamente si è fatto strada dritto verso i nostri joystick.

È stata mantenuta la creazione procedurale delle mappe e praticamente sono presenti la maggior parte delle caratteristiche della versione originale. Possiamo schioccare la frusta, lanciare candelotti di dinamite, arrampicarci, salvare i dispersi nelle caverne, evitare mostri e schivare le trappole.

Certo è più spoglio ma la cura per le animazioni e gli sprite è maniacale. Koller ha racchiuso tutto in 64kb e il

GIUDIZIO FINALE

» Giocabilità 90%

Mappe gigantesche, sempre diverse e una marea di cose da fare, vedere, saltare e colpire. Un titolo sviluppato con grande passione.

» Longevità 90%

Gioco che chiede tanta pratica ma che regala grandi soddisfazioni. Mai ripetitivo.

lavoro è superbo.

Lo possiamo caricare sui nostri C64 per vederlo girare al meglio oppure giocarci in emulazione su VICE o similari.

Lo Spelunky originale è stato realizzato nel 2008 per Windows da Derek Yu e si è dimostrato un titolo vincente per le meccaniche di gameplay e la semplicità di apprendimento. Nel 2009 fu portato su Xbox live arcade e convertito in seguito su PS3, PS Vita, PS4 e Nintendo Switch. Nel 2020, per via del successo, è stato rilasciato il sequel Spelunky 2 per tutte le nuove piattaforme.

C'è da rimanere soddisfatti degli ultimi titoli sviluppati per le nostre retro piattaforme. Anche questa volta ci troviamo di fronte ad un titolo bello da giocare, bello da vedere e innovativo grazie al sistema di generazione mappe procedurale.

Unico appunto la difficoltà. Spelunky è un gioco che va ragionato e non un semplice platform. Può essere piuttosto punitivo se preso sottogamba e richiede logica e ragionamento.

Fiondatevi ad acquistare questo titolo sul sito dello sviluppatore per soli 2 dollari e 99. Ne vale la pena.

di **Roberto Del Mar Pirazzini**

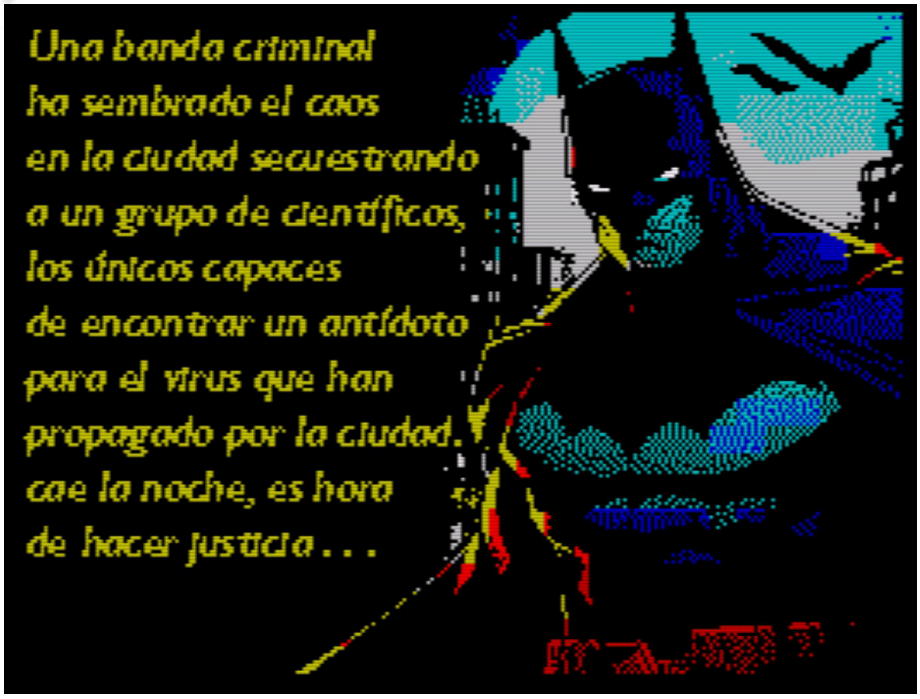
NEW GAME

DARK KNIGHT
SHADOW OF MADNESS

Anno: 2024
Editore/Sviluppatore: Mananuk
Genere: Platform/Action
Piattaforma: ZX Spectrum
Sito web: https://mananuk.itch.io/dark-knight-zxspectrum

"E' l'uomo pipistrello, è Batman... si avvolge nel mantello, è proprio Batman...".
Scommetto che l'avete cantata vero? Pensando alla vostra infanzia o magari a Cristina D'Avena (se avete superato i 40 anni so che avete pensato a lei). Ad ogni modo ho attirato la vostra attenzione e i vostri pensieri per parlarvi di un nuovo gioco per il nostro ZX dedicato al Crociato incappucciato. Lo sviluppatore Mananuk ha realizzato un simpatico action game su 4 livelli divertente che ci ha piacevolmente stupiti.
Un gruppo terroristico ha rilasciato un terribile virus mortale su Gotham City e in più gli scienziati, unici capaci di trovare l'antidoto sono stati rapiti. E quando tutto questo accade chi chiamerai? I Ghostbusters no perché sono un po' anziani e in pensione, ma lui è sempre in caccia e pronto a difendere la città di Gotham.
Il gioco è diviso in quattro livelli diversi. Nel primo livello dovremo saltare da una piattaforma e l'altra liberando gli scienziati rapiti. Ovviamente evitando di morire in modo doloroso per via di trappole e nemici vari.
Il secondo livello è ambientato tra le strade di Gotham ed è un divertente driving game con visuale dall'alto. Nel terzo livello saremo alla caccia di un campione del virus da portare al laboratorio. Il quarto livello è il "randez-vous" finale con tanto di scontro con il boss.
Ma come solo quattro livelli? Si, e meno male! Perché sono decisamente tosti da affrontare e richiedono pratica e numerose partite prima di essere portati a termine.

Il nostro eroe mascherato può eseguire diverse mosse speciali: il doppio salto, lanciare il suo batarang e arrampicarsi sui muri. Sembrano banalità ma sono eseguibili tutte con una serie di



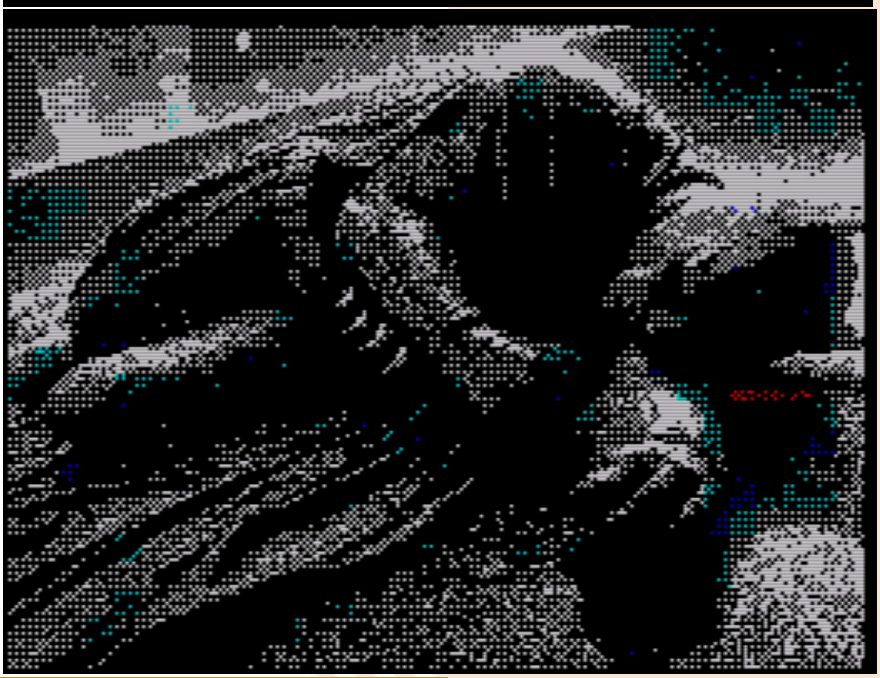
GIUDIZIO FINALE

» Giocabilità 80%
I comandi sembrano semplici da apprendere ma richiedono manualità. I 4 livelli sono davvero ben fatti e divertenti.

» Longevità 80%
Il divertimento è assicurato e, se avete acquistato il gioco, l'espansione vi offre un grado di sfida ancora più elevata.

movimenti dello stick ben fatti.
Il gioco è disponibile in versione digitale gratuita ma, se donate qualche spicciolo di mancia, vi verrà inviata l'espansione con Robin, un nuovo veicolo e un livello di difficoltà maggiore.
Dark Knight è un buon titolo. Uno di quelli da giocare con impegno e da apprezzare per qualità e per idee. Forse non sarà la miglior esperienza ludica su ZX ma merita di essere provato, soprattutto se si è fan del personaggio creato da Bob Kane per la DC Comics.

di **Giampaolo Moraschi**

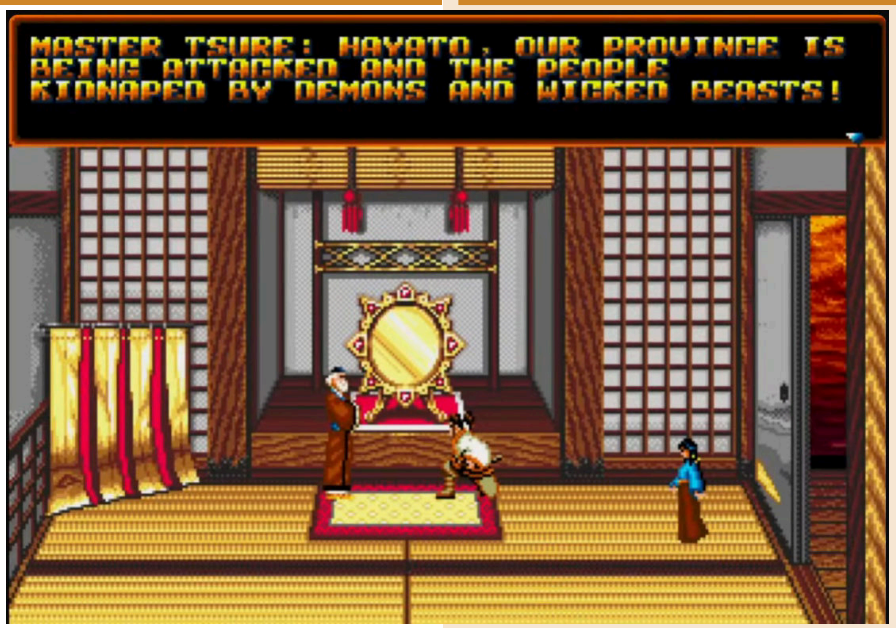


NEW GAME

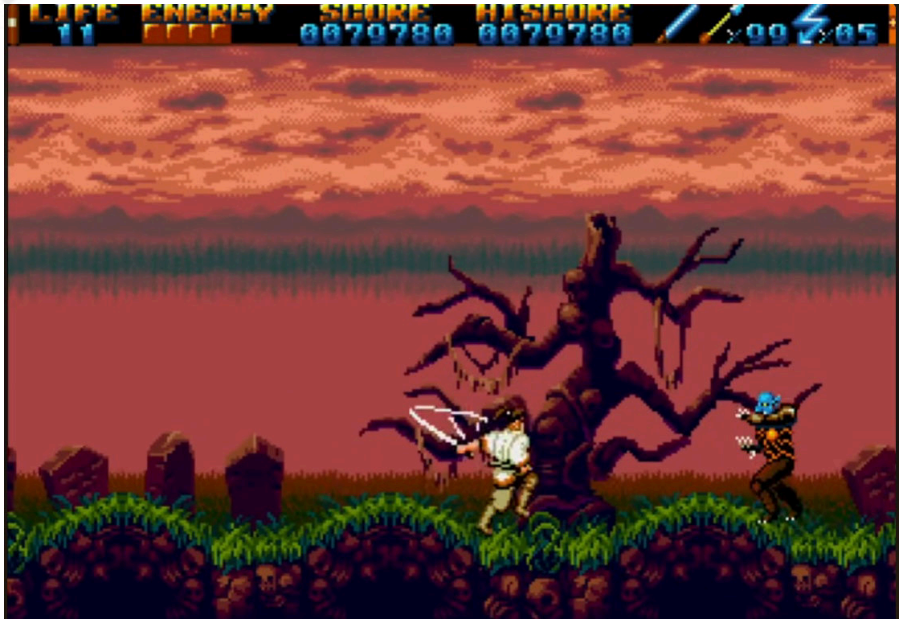
HAYATO'S JOURNEY

Anno: 2024
Editore/Sviluppatore: Master Linkuei
Genere: Action/Platform
Piattaforma: Sega Megadrive
Sito web: https://master-linkuei.itch.io/hayatos-journey

Hayato's Journey è un chiaro omaggio ad un genere molto amato sulle piattaforme Sega 8/16 bit, quello dei platform/action in stile nipponico.
Un omaggio non ufficiale a Kenseiden, un classico di culto per Sega Master System ma che di Kenseiden presenta poco se non il protagonista. Invece deve molto alla saga di Shinobi, con una struttura di gioco a livelli crescente, salvataggio di prigionieri e boss finale.
Hayato, il protagonista, è il discendente del Grande Hayato, un membro leggendario di una famiglia di Draghi che combatté contro il malvagio Warlock secoli fa.
Il malvagio stregone Yonensai in cerca di vendetta decide di porre fine alla discendenza del clan Hayato uccidendo e rapendo la famiglia dei Draghi.
Il protagonista, equipaggiato dalla sacra spada del Re Drago, si getterà alla ricerca del malvagio stregone attraversando numerosi scenari pieni zeppi di creature malvagie, spiriti, mostri ed entità maligne.
Il titolo è sviluppato con il versatile Scorpion Engine e si tratta del primo titolo per Sega Megadrive completato con questo motore.
Prende in prestito da diversi videogames del passato alcuni sprite e fondali e si sviluppa nel classico platform game con elementi d'azione: distruggi il nemico, recupera i power up, sconfiggi il boss… e così via.
Per variare alcuni livelli ci vedranno sulla groppa di un cavallo, sopra una barca e su una tavola di legno. Un modo simpatico per rinnovare l'azione che spesso potrebbe apparire monotona.
Come vi dicevo è un omaggio a



Kenseiden, un titolo che mescolava elementi d'azione all'esplorazione dove potevamo potenziare il personaggio per poter proseguire.
In Hayato's Journey tutto questo è stato tolto e l'azione ricorda proprio Shinobi e i suoi seguiti. Una scelta che forse farà storcere il naso ai cultori del titolo originario per Sega Master System.



GIUDIZIO FINALE

» Giocabilità 75%

Tanta azione di gioco e diversi power up ma una strana scelta di design dei livelli e un bilanciamento non corretto dei power up e dei boss finali.

» Longevità 70%

Senza i power up è un titolo difficilotto... con i power up al massimo lo finirete in poco tempo.

Dal punto di vista tecnico è un titolo particolare. Abbiamo alcuni elementi interessanti come diversi fondali con effetti di parallasse, ottimo uso dei colori e, di contro, la scelta di utilizzare spesso elementi da altri videogiochi. Scelta furba e utile ma che in realtà crea più confusione e non armonizza l'aspetto estetico.

Il sonoro invece presente degli effetti campionati vocali ben fatti e rari su Megadrive per qualità e pulizia.

Interessanti anche le colonne sonore.

È un gioco vario e i livelli sono tanti ma soffre in termini di giocabilità. Senza i power up e il corretto uso delle armi opzionali o della smart bomb è difficile proseguire. Paradossalmente con la spada potenziata, l'arco e la già citata smart, il gioco risulterà piuttosto semplice. Avrei bilanciato meglio.

Avrei anche aggiunto un maggior numero di power up per rendere l'azione più varia.

Concludendo è un buon titolo che merita di essere provato e gustato se si amano i classici come Shinobi.

di **Giampaolo Moraschi**

NEW GAME

SUPER DASSALO LAND

Anno: 2024
Editore/Sviluppatore: Gumpy Fuction
Genere: Platform
Piattaforma: Game Boy color
Sito web: https://gumpyfunction.itch.io/super-dassalo-land?download

Tommy Dassalo è un artista di Stand Up comedy realmente esistente. Questo strambo personaggio irriverente australiano è il protagonista del platform per GBC che vi presento oggi.

Tommy è pronto all'evento culturale del millennio: la registrazione del suo speciale di stand-up comedy "SCAM ARTIST".

Ma proprio mentre tutto sta per partire ecco apparire il terribile (?) fantasma Jiko che intrappola il nostro occhialuto eroe tra le pagine del suo quaderno di battute... assieme al suo cane Kewpie.

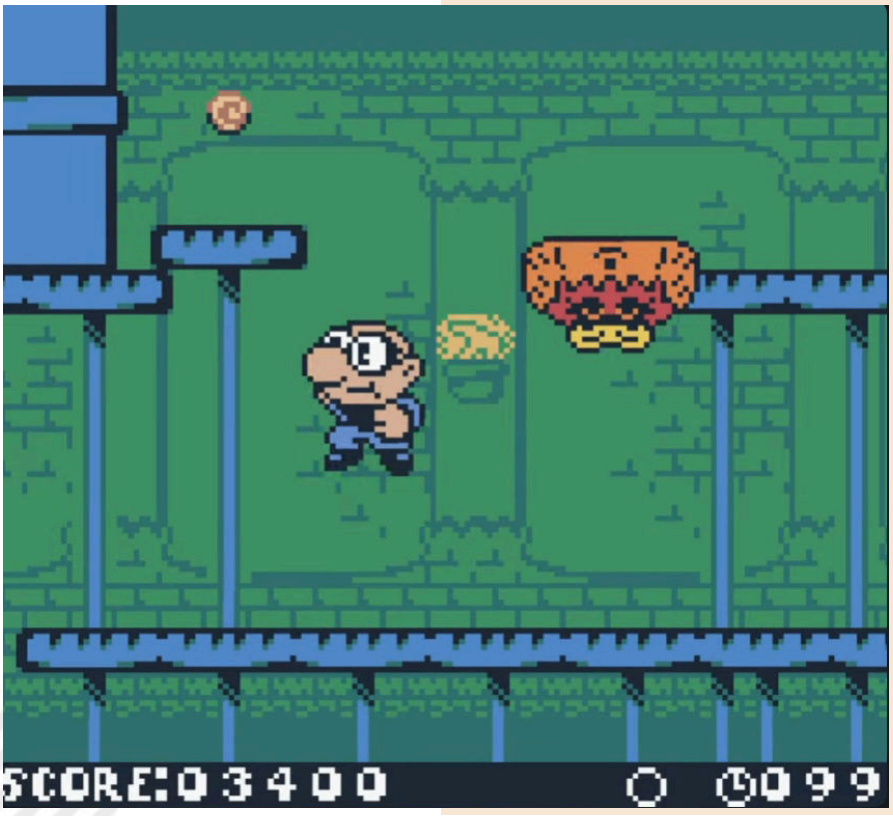
Qui Kewpie sarà il mentore di Dassalo. Un mentore molto particolare: parla ed è molto irritabile, volgare e non sopporta la stupidità del suo padrone. Noi giocatori ovviamente saremo gli aiutanti umani di Dassalo attraverso 3 mondi pieni di mostri bislacchi, trappole, salti, sconcezze e follia e saremo equipaggiati di super frustra microfono e di abilità di salto di vario tipo.

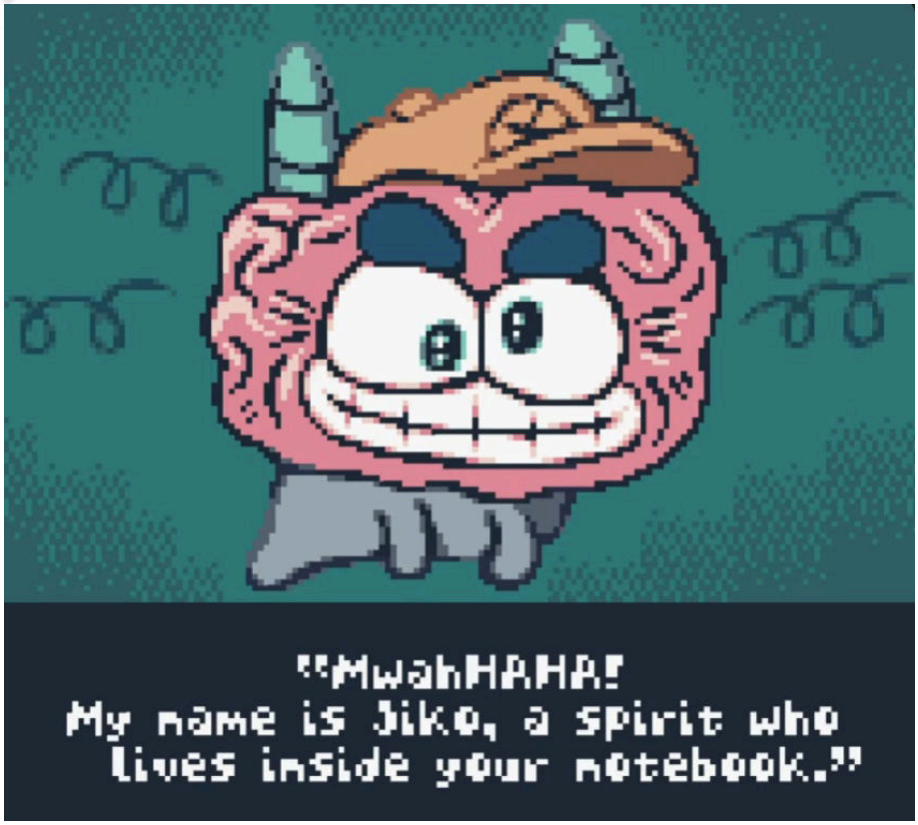
È un platform pieno di umorismo volgare che potrebbe non essere adatto ai più piccoli, ma che mi ha fatto morire dal ridere per le follie del suo protagonista/ispiratore.

Nulla di nuovo per questo genere di giochi su Game Boy, ma è un titolo gratuito che si lascia giocare per qualche tempo.

Il vero difetto del gioco è la sua terrificante semplicità. Arrivare al terzo mondo è un attimo e si rimane con l'amaro in bocca.

Si perché il protagonista è quanto di





meno eroico si possa pensare. Paffuto, calvo e con due grandi occhiali tondi continuamente vessato dal suo cane che cerca di non perdere la pazienza mentre spiega cosa deve fare al nostro strambo eroe.

Il gioco è stato seguito dal vero Tommy Dassalo e quindi risente della sua comicità ai limiti del corretto.

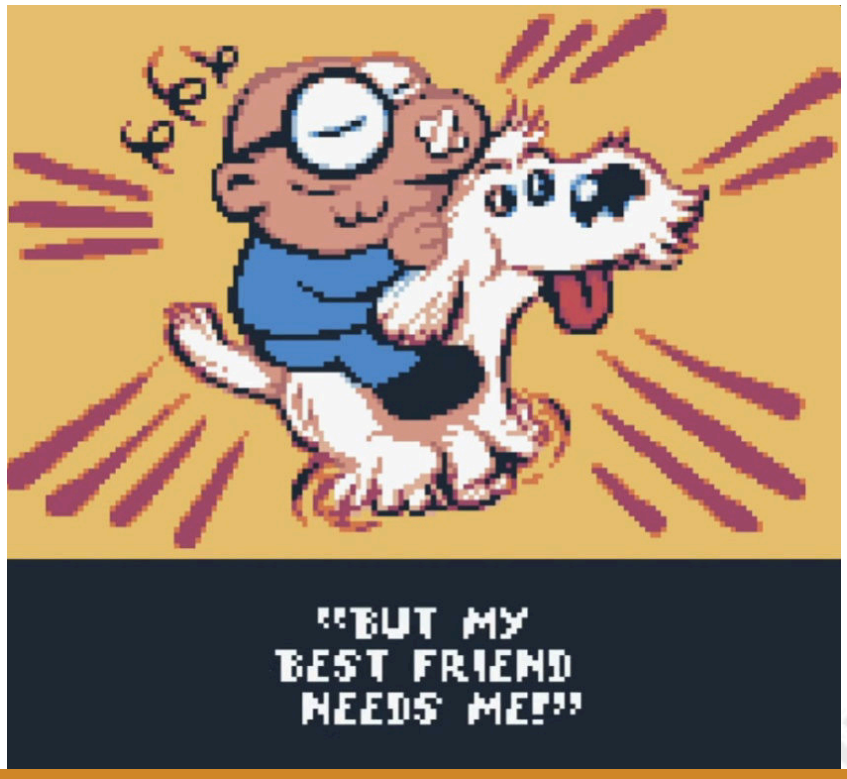
Tecnicamente è gradevole ma non raggiunge mai picchi di eccellenza. Avrei elaborato un po' meglio alcuni fondali che rischiano di confondersi con piattaforme o salti ciechi.

Semplice da apprendere e troppo, troppo, troppo facile.

Consolazione? È gratis e si può anche giocare da browser.

P.s. La battaglia finale a suon di pupù è devastante!

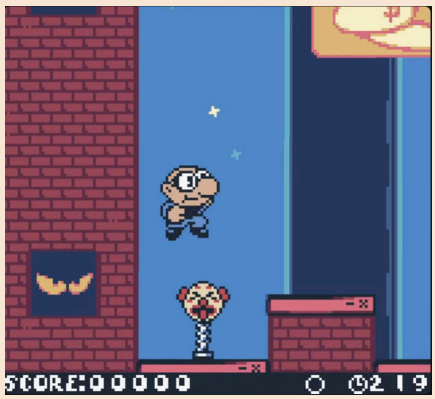
di **Marta Rossmann**



GIUDIZIO FINALE

» Giocabilità 70%
Un platform con tutto al suo posto, ma piuttosto simile ad altri mille simili.

» Longevità 50%
L'umorismo folle mi ha colpita, ma il gioco è troppo semplice da terminare.



NEW GAME

BLACK & WHITE

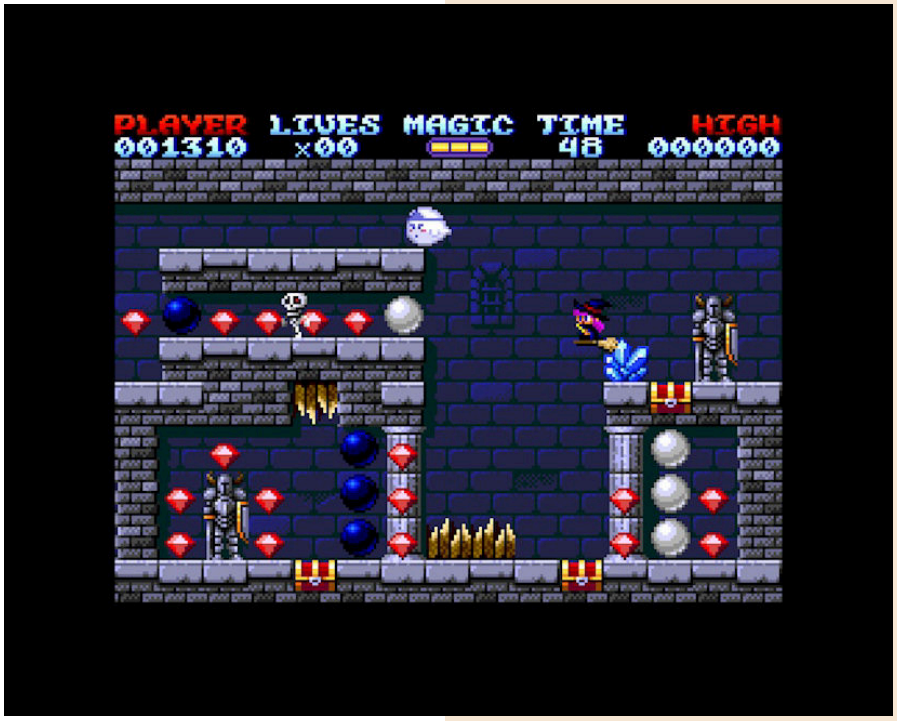
Anno: 2024
Editore/Sviluppatore: Pat Morita Team
Genere: Platform
Piattaforma: ZX Spectrum NEXT o clone
Sito web: https://patmorita.itch.io/black-and-white-spectrum-next

La formula vincente per un buon gioco? La semplicità. Pochi fronzoli e poche cose da sapere possono rendere il titolo perfetto ed è questo il caso di Black & White.

Un gioco semplice nelle sue meccaniche e perfetto nel divertimento. Un platform dei più classici che ci vede nei panni delle streghette gemelle Candel e Alice.

Queste due biricchine hanno l'ingrato compito di rimettere in ordine il regno di Kalela e dovranno farlo recuperando i diamanti sparsi in 6 livelli da esplorare: le High Mountain, la foresta oscura, Coco beach, il castello nero, le caverne e le rovine Azteche.

Sfortunatamente, un pasticcio le ha fatte fondere in un'unica entità. Cosa significa questo? Che le due maghette sono fuse assieme ed è necessario utilizzarle entrambe per poter raccogliere tutti i diamanti sparsi nei livelli.



Lo switch dei "colori" avviene quando una delle due esaurisce il "mana" magico. Aspetto interessante che stabilisce il modo per portare a termine il livello. Un sistema curioso ma molto intrigante, che dona al giocatore un po' di pepe in più per affrontare il mondo di gioco.

E' bene ricordare che le nostre eroine non possono saltare, ma solo muoversi nel livello e che le loro bacchette possono solo stordire gli strambi avversari.

Vedete? Ci sono tutte le meccaniche per rendere avvincente l'avventura! Inutile dirvi che i livelli offrono un approccio differente alla raccolta delle gemme preziose, che sono presenti diversi nemici dall'aspetto "kawaii" ma letali e che, per aiutarci, sono stati inseriti oggetti preziosi e il tempo extra. Giusto... il tempo! Si perché il timer per ripulire il livello, diminuisce sempre

di più e sarà necessario fare in fretta. Ho seguito per diversi mesi lo sviluppo del titolo e non ne sono rimasto deluso. Penso davvero che lo sviluppo di giochi su ZX Spectrum NEXT sia una porta dimensionale; un collante tra i vecchi titoli degli anni '80 e la new wave dei giochi in stile retrò.

Aspetto grafico meraviglioso. Coloratissimo, animato perfettamente e con tantissima varietà nei paesaggi e negli sprite. Sonoro perfetto con una bella musica d'accompagnamento in gioco e effetti sonori perfetti.

Si fa tutto con 1 tasto e con il joystick e il gioco è una goduria con i suoi due livelli di difficoltà (normale e speciale). Il titolo nasce per ZX Spectrum Next o per uno dei suoi cloni (come l'N-GO) ma funziona alla grandissima su dispositivi FPGA con i core del Next o in emulazione (consigliato l'emulatore CSPect per Windows).

Costa solo 5, 99 dollari e li merita tutti (e anche qualcosa di più) e sicuramente sarà da tenere d'occhio una futura release in versione "fisica".

Per quanto mi riguarda questo titolo (assieme ad Alien Neoplasma 1 e 2) è una delle Killer Application per il NEXT.

di **Carlo Nithaiah Del Mar Pirazzini**

GIUDIZIO FINALE

» Giocabilità 95%

Si gioca in modo semplice ma con una meccanica intrigante nello switch delle maghette e nell'approccio dei livelli.

» Longevità 95%

Ci sono 6 mondi con 60 livelli da esplorare e due differenti difficoltà. Bestiale!

NEW GAME

DATA MAN

Anno: 2024
Sviluppatore: Broke Studio/ Darkbits
Genere: Shoot em up
Piattaforma: Nintendo NES, Windows, macOS
Sito web: https://darkbitsgames.itch.io/data-man

Allarme virus! Allarme virus!
Master e i suoi scagnozzi hanno invaso il sistema e non si fermeranno finché ogni dato non sarà corrotto.
Ed è qui che entrano in gioco i Data Men, eroi miniaturizzati con il compito di ripulire le PCB dai terribili parassiti che vogliono mangiarle.
Lo scopo è semplice: blastare ogni singolo mostriciattolo insettoso che tenti di divorare le CPU delle nostre amate schede.
I nostri Data Men sono dotati di pistola laser e di granate. La pistola ha un tempo di caricamento ed è potenziabile e le granate sono a numero limitato, ma è possibile recuperarne altre.
Il gioco è un bel top down shooter che ricorda Smash Tv ma ambientato su PCB di ogni tipo, forma e colore.
Distruggere questi mostriciattoli ci ha entusiasmato molto e dobbiamo segnalare questo titolo tra i migliori giochi per il Nintendo 8 Bit di tutto questo 2024.
Ci sono 20 livellli di gioco e 4 boss fight con una difficoltà crescente. Ogni livello si può terminare dopo tre wave di mostri che sbucano fuori da ogni parte.
E' possibile giocare in doppio per una giocabilità incredibile.
Questo titolo è nato per tre diverse piattaforme. Acquistando la digital rom per NES si avranno disponibili anche gli installer per Windows e Mac.

I ragazzi di Broke Studio hanno realizzato anche una bella cartuccia da inserire nei nostri Nintendo che comprende anche un bel Manuale.
Data Man è un gioco ben fatto che mescola un comparto tecnico minimalista ma gradevolissimo e una

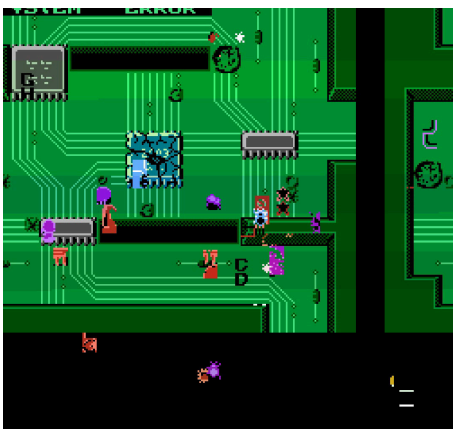
GIUDIZIO FINALE

» Giocabilità 95%
Semplice e con un gameplay ben congeniato.

» Longevità 95%
Bellissimo in singolo ma in doppio è micidiale! Livelli dalla difficoltà crescente e un livello di difficoltà supplementare tutto da scoprire.

giocabilità accattivante che porta a voler vedere sempre di più.
La difficoltà è ben bilanciata anche se non è un titolo da prendere sotto gamba: è necessaria molta manualità per affrontare i numerosi livelli di gioco.
Simpatica la possibilità di non morire mai ma di essere rallentati e bellissima la sequenza di distruzione della PCB con tanto di crash di sistema.
Se possedete un NES fisico comprante al volo questa cartuccia, se avete l'everdrive o volete giocarci in emulazione cosa aspettate a comprare la digital edition?
Capolavoro!

di **Carlo Nithaiah Del Mar Pirazzini**



NEW GAME

RUFF IN TROUBLE

Anno:1992/2024
Editore/Sviluppatore: Shanti77
Genere: Platform
Piattaforma: Atari 8bit
Sito web: https://forums.atariage.com/topic/374485-ruff-in-trouble/

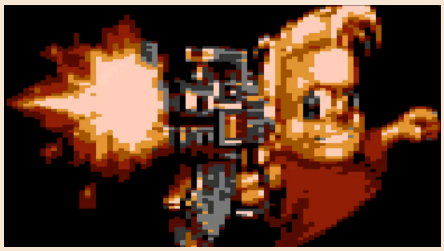
Si è da poco conclusa l'ABBUC competition 2024 dedicata alle macchine Atari 8bit. Un livello impressionante di titoli in gara e tutti di grandissimo livello.

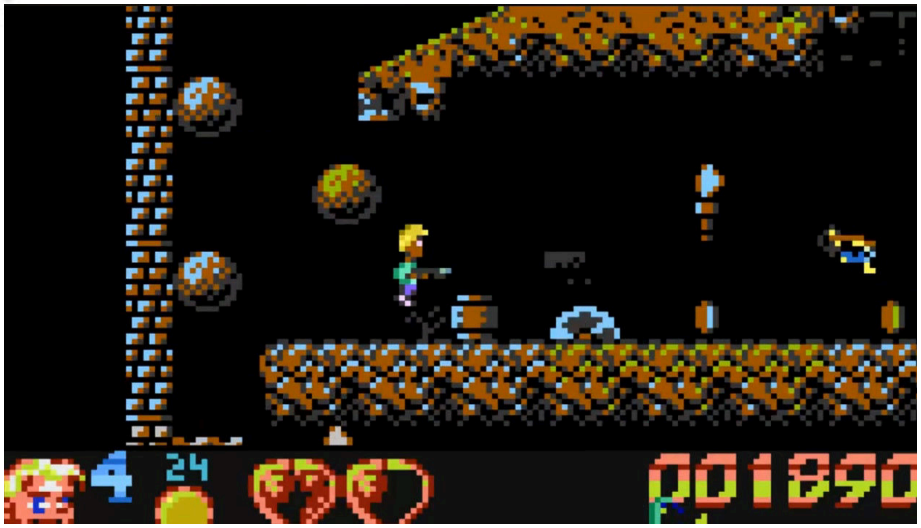
Il vincitore è risultato questo Ruff in Trouble realizzato da Shanti77. Si tratta di un platform game che omaggia il bellissimo Ruff 'n Tumble su Amiga, uno degli ultimi titoli per la piattaforma Commodore nell'era passata.

Il gioco vede Ruff cercare di uscire incolume da un inospitale mondo alieno pieno di pericoli. Per uscire dal livello sarà necessario recuperare delle sfere sparse in tutto il perimetro della mappa. Una volta recuperate si aprirà il portone e si accederà al livello successivo.

Il gioco richiede Atari con 128 KB di ram per funzionare, supporta i joystick a doppio pulsante e gira senza problemi in modalità Pal (su NTSC è compatibile ma con qualche rallentamento).

Il lavoro fatto da Shanti sull'aspetto tecnico è eccellente e mette in mostra





GIUDIZIO FINALE

» Giocabilità 92%
Livelli grandi e pieni di dettagli mescolati ad un buon uso dei comandi di gioco e alla presenza di un grado di sfida fin da subito serrato.

» Longevità 90%
Richiede pratica e disciplina ma è divertentissimo!

una grafica dettagliata, colorata e animata con cura.

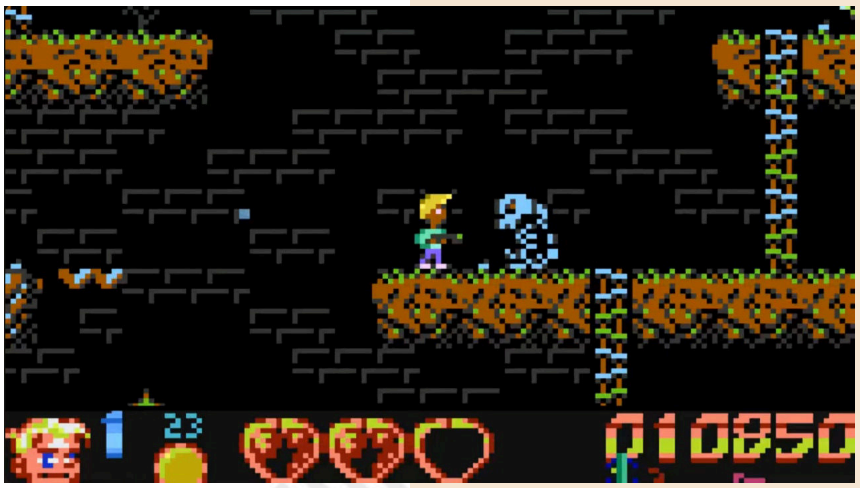
Le mappe dei livelli si lasciano esplorare e ci si perde tempo ad apprezzare i dettagli scelti.

A parte la grafica incredibile, il livello di difficoltà è in realtà impegnativo come lo era Ruff 'n Tumble su Amiga. Ci sono un bel po' di trappole da affrontare, e i nemici vanno approcciati con intelligenza per evitare morte prematura.

Altro punto di forza è il modo come approcciare il livello. E' un titolo multidirezionale che permette di esplorare con cura ogni anfratto possibile.

La sfida per vincere questa edizione dell'ABBUC era decisamente elevata ma Shanti ci ha regalato un titolo davvero ben fatto che merita di essere giocato e supportato.

di **Giampaolo Moraschi**



NEW GAME

SUPER MARIO BROS MINI

Anno: 2024
Sviluppatore: Mico27
Genere: Platform
Piattaforma: Game Boy Color
Sito web: https://mico27.itch.io/super-mario-bros-mini

Dal 1985 ad oggi sappiamo tutto di Super Mario Bros, primo capitolo della saga "Super" del noto idraulico italo-nipponico di Nintendo.

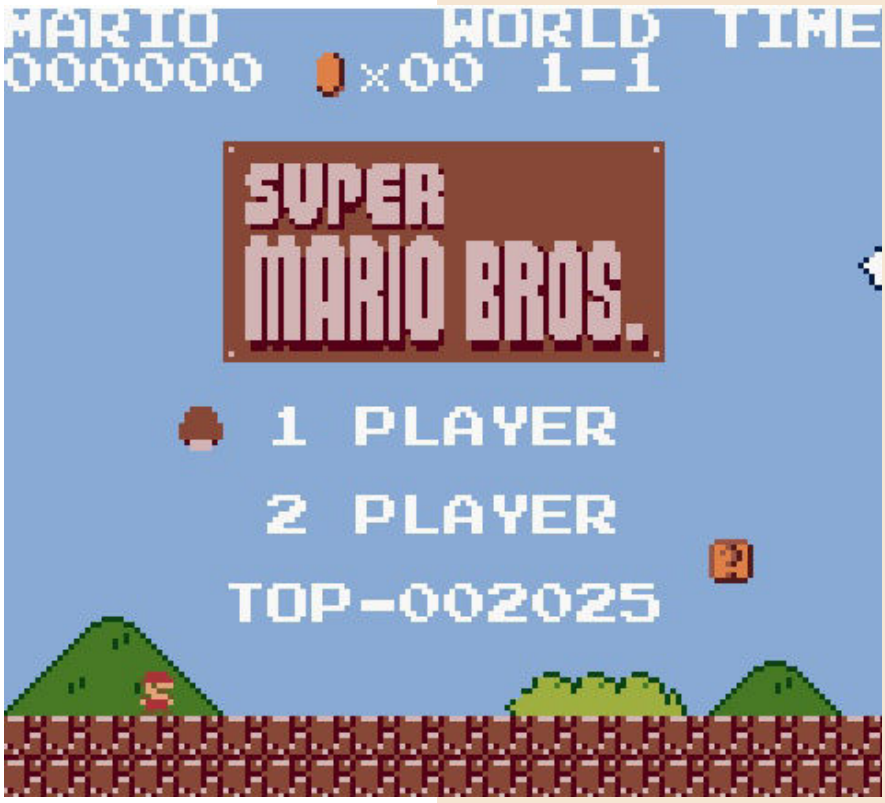
Abbiamo visto l'eroe baffuto correre in coloratissimi mondi per recuperare monete e salvare la principessa Peach dalle grinfie di Bowser, acerrimo nemico di tutta la saga.

Un successo incredibile per la Nintendo e la vera Killer Application per il NES al suo lancio negli Stati Uniti e in Europa (il NES in Giappone era già presente da due anni). Dal mitico primo capitolo abbiamo visto seguiti in tutte le console di casa Nintendo, tutti splendidi esempi di gameplay.

Ma il fascino del primo Super Mario Bros è immutabile. Abbiamo visto Speed Runner cimentarsi nel completare più velocemente possibile il gioco, altri giocatori scoprirne tutti i più incredibili segreti... abbiamo persino visto la bellissima conversione per Commodore 64 e la demo di quella per Intellivision (entrambe "oscurate" dalla Nintendo).

Mancava di vedere questa versione per Game Boy Color, realizzata da Mico27 con il GBStudio.

C'è tutto il gioco originale dentro questo piccolissimo file pixeloso e non solo! Mico27 ci ha inserito una nuova modalità di gioco con 8 nuovi mondi aggiuntivi, con nuovi temi grafici, nuovi livelli, un nuovo



potenziamento, alcuni “amici”, tanti nuovi boss e… molto altro ancora.
Come se tutto questo non bastasse ha inserito una modalità “Random” che permette di affrontare l’avventura in modo davvero alternativo.
Tutto questo lo si può ottenere premendo B nel menù principale e selezionare ciò che si vuole affrontare.
Il GBStudio è una build potente e ben sviluppata che permette agli addetti ai lavori di realizzare titoli, demo o esperimenti in modo semplice ma con tantissima cura per la qualità finale.
Qui il lavoro è davvero pregevole. C’è tutto quello presente nel gioco originale. Sonoro perfetto (nei limiti del GBC) e la mini grafica “kawaii” curatissima.
Uno dei prodotti da considerare, testare e provare prima che la Nintendo lo cancelli dalla faccia della terra (speriamo di no!!!).
Per finire e per i curiosi vi segnalo che lo sviluppatore ha reso disponibili i codici sorgente nella propria pagina github al seguente indirizzo: https://github.com/Mico27/SuperMarioBrosMini.

Grande titolo per GBC.

di **Carlo Nithaiah Del Mar Pirazzini**

GIUDIZIO FINALE

» Giocabilità 95%
Super Mario Bros possiede meccaniche di gameplay perfette anche nel 2024 e anche su GB Color. Il tutto arricchito da due modalità supplementari.

» Longevità 95%
C’è tutto il gioco originale (warp zone comprese), 8 nuovi mondi e la versione random. Volevate ancora qualcos’altro?

NEW GAME

SUPER MARIO ECLIPSE

Anno: 2024
Sviluppatore: Nintendo, Gamebanana
Genere: Platform game
Piattaforma: Nintendo GameCube
Sito web: https://gamebanana.com/mods/536309

Sunshine è sempre stato uno dei giochi che mi ha affascinata di più su GameCube. Per la sua storia di sviluppo e per i contenuti e per il suo difficilissimo compito di eguagliare il successo di Mario 64.

Quindi giocare questa hack è stata un'esperienza magica.

A dire il vero, non sapevo che mi aspettasse quando l'ho scaricata. Ho prima guardato il trailer e ho pensato che fosse una campagna completamente nuova, ma è stata una sorpresa gradita scoprire che il gioco è una rivisitazione del classico con alcuni nuovi elementi interessanti e con idee che dovevano essere implementate nell'originale.

I cambiamenti nei movimenti sono la novità più grande con l'aggiunta del salto in lungo e del salto a terra (direttamente da 3D World). Queste due nuove mosse aiutano davvero molto nel gioco e nelle missioni. È soddisfacente concatenare queste mosse a tutto il resto del set che era già bellissimo.

Un'altra cosa che aiuta immensamente il flusso di gioco è che non ti butta fuori dai livelli una volta completata una missione. Cavolo! Nel gioco originale ti buttava fuori anche quando prendevi le 100 monete shine! Ora, a seconda di quello che accade nel

livello potrebbe riportarti alla schermata di selezione missione o semplicemente lasciarti continuare a girovagare e cercare se ci sono più Shine disponibili da recuperare. C'è anche un'opzione di ripristino rapido se si vuole cambiare missione nel livello in cui ci si trova, altra manna dal cielo.

Anche le monete blu sono più facili da collezionare, altro aspetto controverso nel Sunshine originale.

GIUDIZIO FINALE

» Giocabilità 92%

Un titolo che migliora tantissimo il precedente e bistrattato Sunshine, aggiungendo una marea di cose in più e tutte molto divertenti.

» Longevità 85%

Il problema di frame rate in alcuni punti è davvero molto fastidioso ma il titolo promette ore e ore di gioco e collezioni da sbloccare.

I nuovi livelli, che erano stati scartati nel titolo originale, sembrano appartenere al gioco con tutto ciò che riguarda il modo in cui sono stati progettati. Si adeguano bene al level design complessivo e alla dinamica di gioco. Quattro livelli con 8 missioni e tante cose da raccogliere, lavoro eccellente.

Eclipse si è rivelato ricco di cose da fare, da raccogliere, di sfide nuove, di finali segreti incredibili e di una quantità di cose post-gioco da far paura.

Forse il punto dolente di Eclipse è la realizzazione tecnica a tratti barcollante. In alcuni livelli il framerate si abbassa in modo micidiale e fastidioso. Sicuramente sarà migliorato con le prossime release che gli sviluppatori hanno promesso nella loro pagina. Tutto sommato il resto regge e il gioco scorre nonostante questo problema.

Super Mario Eclipse è una delle migliori Hack a cui abbia mai giocato, forse la migliore. È un lavoro di un team che ha compreso al 100% la filosofia del gioco originale e ne ha catturato l'essenza in modo (quasi) impeccabile con tutti i nuovissimi contenuti aggiuntivi

di **Marta Rossmann**

NEW GAME

TONY MONTEZUMA'S GOLD

Anno: 2024
Editore/Sviluppatore: Monochrome Productions
Genere: Platform
Piattaforma: Commodore 64, Amiga, Atari 8bit
Sito web: https://monochrome-productions.itch.io/

Un'antica maschera leggendaria... Una fitta giungla misteriosa... Templi maledetti... Un eroe senza macchia e senza paura... un joystick... e noi giocatori!

In questa estate italiana ci getteremo alla ricerca della maschera di Montezuma. Non un semplice artefatto, ma un monile dai poteri misteriosi.

Un bel platform vecchio stile che omaggia i grandi classici del passato e in particolar modo Montezuma's Revenge della Parker Brothers.

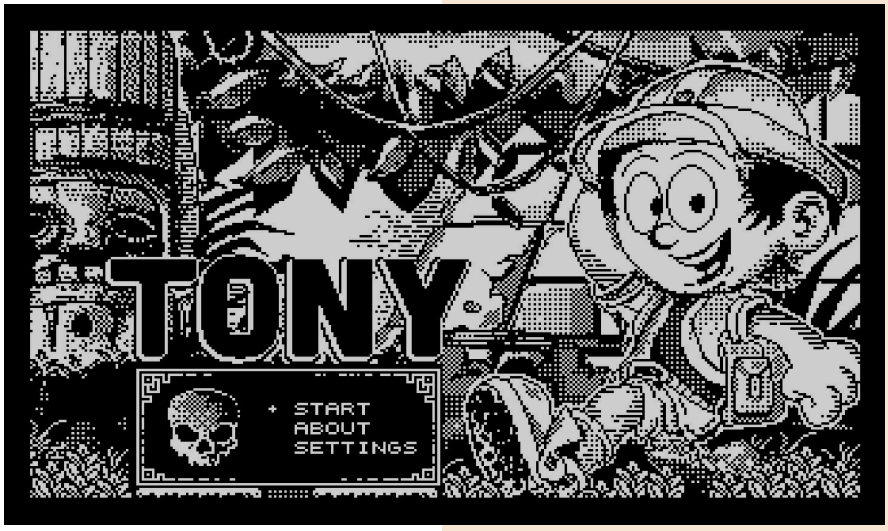
Omaggia anche la figura dell'esploratore e regista polacco Tony Halik, noto per le sue avventure attraverso tutti i continenti.

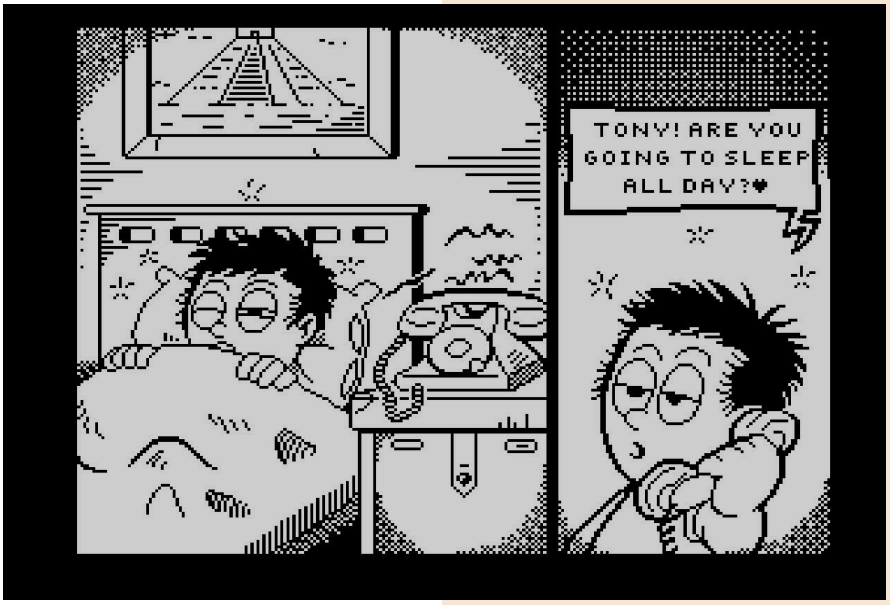
Nei panni di Tony, il giocatore attraverserà il castello di Montezuma, pieno di trappole e creature letali. Impresa complessa e piena di salti "pixel per pixel" e di oggetti da raccogliere.

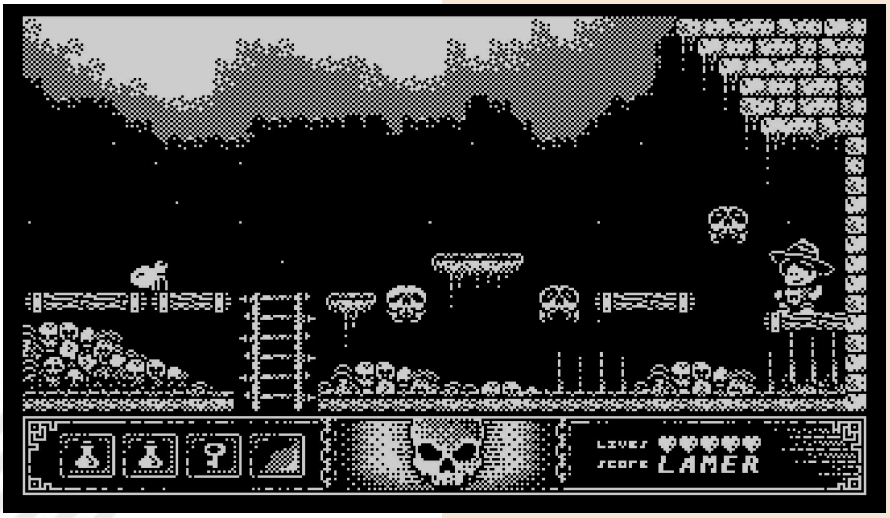
Fondamentali per la sanità mentale e per la sopravvivenza saranno i gioielli sparsi lungo i livelli, le pozioni che ci permetteranno di eliminare i nemici, le maschere dorate e le chiavi. Questi ultimi due oggetti fondamentali per proseguire nel livello.

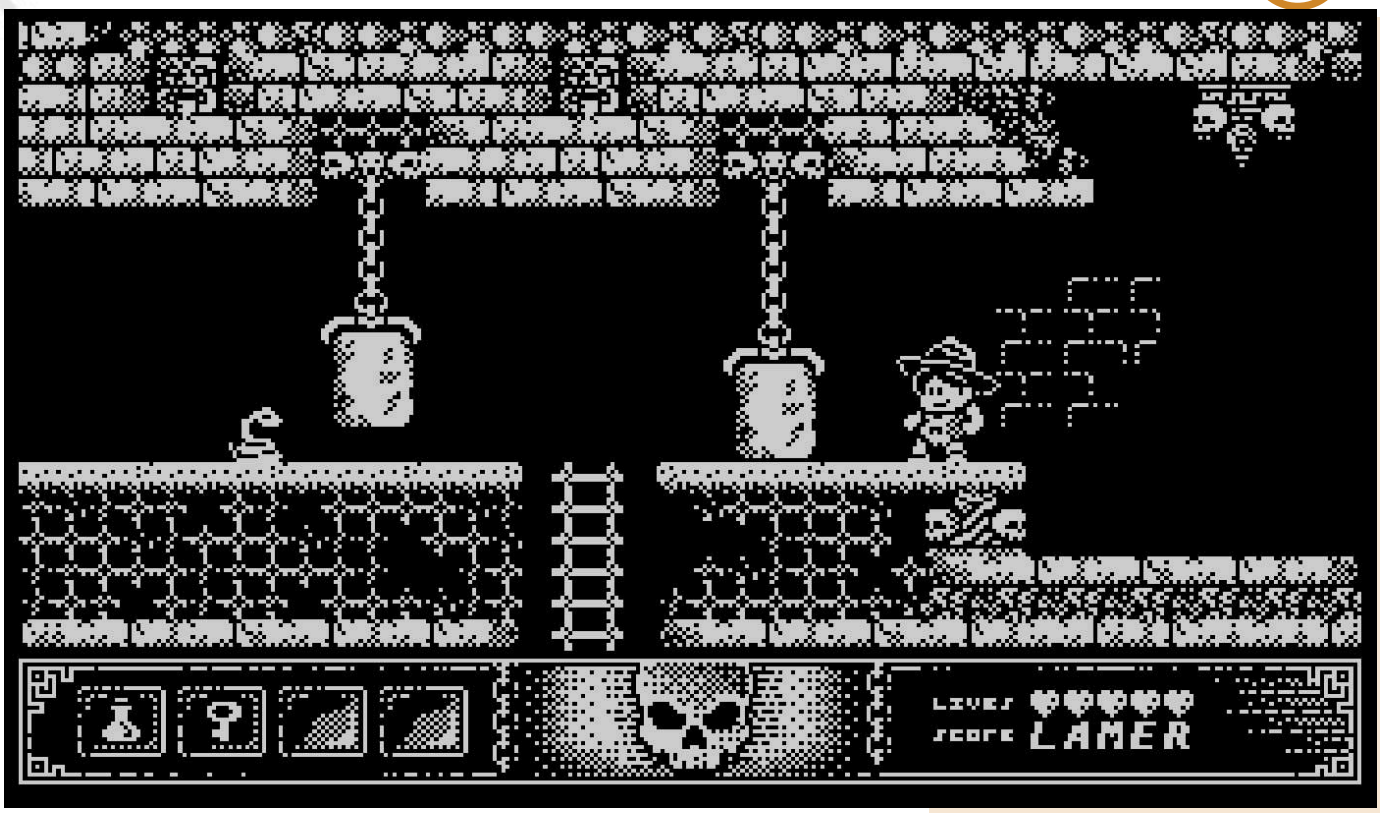
Non sarà facile trasportarle però! L'inventario a disposizione del giocatore è di soli quattro slot e sarà basilare scegliere cosa infilarci dentro.

In ultimo, il giocatore dovrà osservare con attenzione le numerose e (ben) nascoste stanze segrete, piene di tesori aggiuntivi. Gli sviluppatori le hanno inserite nei posti più impensabili.









Tony Montezuma's Gold è giunto in tre belle versioni contraddistinte da una sola ed unica caratteristica: la grafica monocromatica.

Scelta di stile particolare, ma molto azzeccata che rimanda sempre al periodo della "Golden Age" dei videogiochi.

Nonostante il bianco e nero, la grafica in tutte e tre le versioni (Amiga, Commodore e Atari 8bit) e molto ma molto gradevole.

Ben animata e dettagliata ma con leggere differenze ovviamente in risoluzione e animazioni. In tutti e tre i casi però il gioco ha davvero un bel aspetto e gira che è una meraviglia.

Bello anche il sonoro in modo particolare su Commodore 64 e su Atari. La colonna sonora chiptune l'ho trovata molto coinvolgente mentre non ho gradito gli effetti sonori della versione per Amiga.

Molto carina anche la presentazione grafica in stile "fumetto" che narra l'avventura del povero Tony dalla camera da letto alle foreste amazoniche.

Arriviamo all'aspetto più giocabile.

I livelli di gioco sono ben strutturati in tutte e tre le versioni e identici. È il tipico gioco che richiede memoria e quel tipo di gioco "salto per salto perfetto" non sempre alla mano per i giocatori casuali.

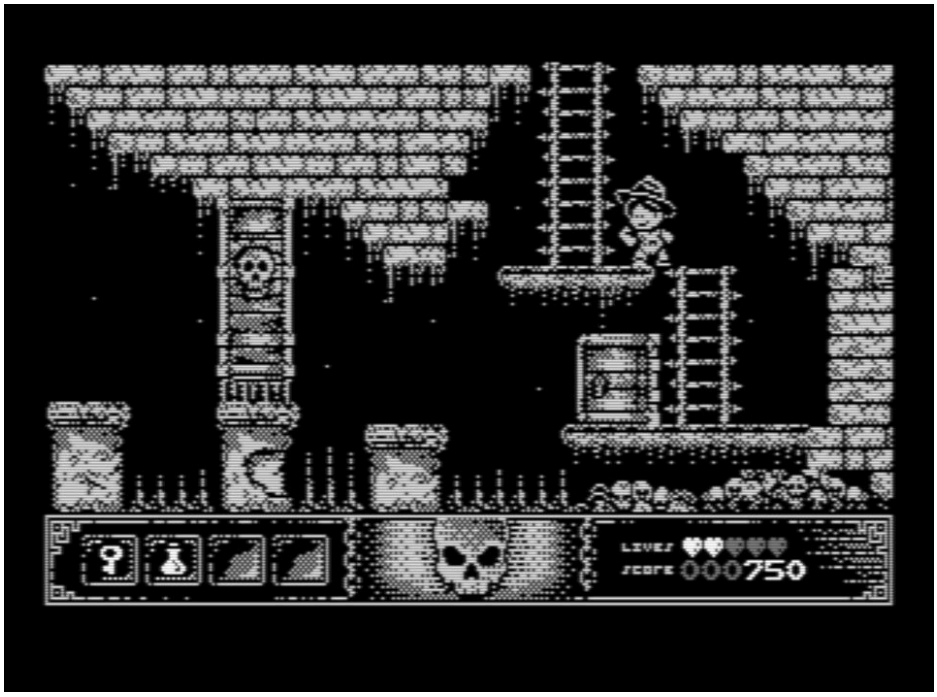
Tra le tre versioni ho trovato più debole quella Amiga. Identica in tutto ma più "ostica" nei salti e nella dinamica di gioco del personaggio che sembra più ingessato rispetto alle altre due versioni.

In ogni caso è un gioco tosto che richiede diverse ore per essere padroneggiato ed una buona memoria.

Resta però un grande prodotto in tutte

Fig. 1,2,3,4 - Immagini tratte dalla versione Amiga. La schermata introduttiva, un frame della simpatica introduzione e due immagini del gioco in esecuzione. L'estetica monocromatica è splendida!

Fig. 5,6 (pagina successiva) - Immagini tratte dalla versione Commodre 64. Schermata introduttiva e una fase del gioco. Stessa atmosfera della versione Amiga.



GIUDIZIO FINALE

» Giocabilità 85%

Una media delle versioni. I livelli sono ben sviluppati ed il sistema di gioco collaudato e godibile. Su Amiga il protagonista è meno “agile” che nelle altre versioni e alcuni salti non sono proprio perfetti (questo limita molto la godibilità).

» Longevità 90%

Ostico per i giocatori “casuali” ma con quel bel gusto dei favoli anni 80. Salti su salti da effettuare memorizzando il livello.

e tre le versioni e di sicuro merita la sufficienza piena e anche qualcosa di più. Su Commodore è in formato disco (diviso in due file) e funziona in real hardware e in emulazione.
Anche su Atari nessun problema in real hardware o su Altirra.
La versione Amiga richiede un kickstar 1.2, 68000, 1 mb di Fast e 0,5 di Chip, gira su OCS/ECS/AGA e ovviamente in emulazione.
Tutte e tre le versioni avranno diverse release in versione fisica su Disco (C64 e AMIGA) e Cartuccia su ATARI con manuali, gadget e tanto altro.
Complimenti davvero al team di sviluppo per aver realizzato un prodotto di questo genere nel 2024. Bravi!
Carlo Nithaiah Del Mar Pirazzini

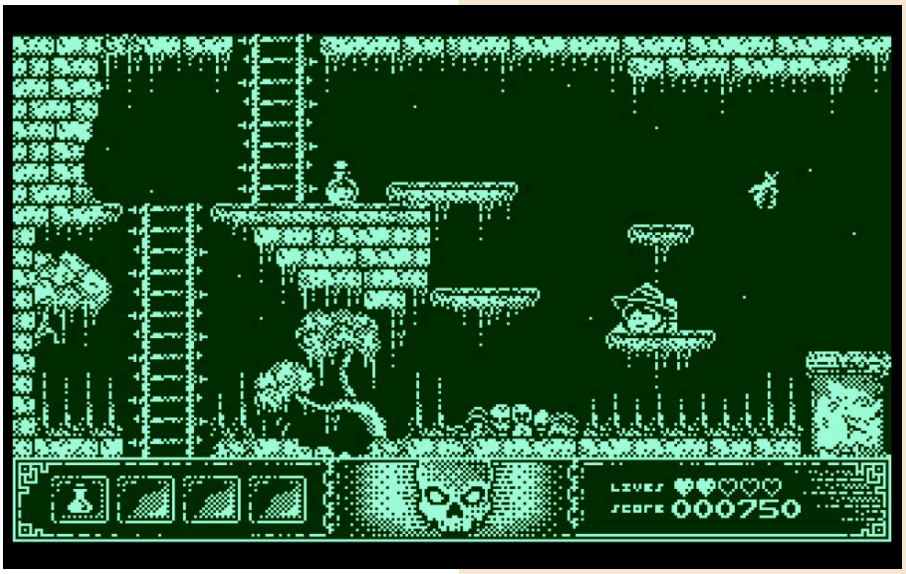
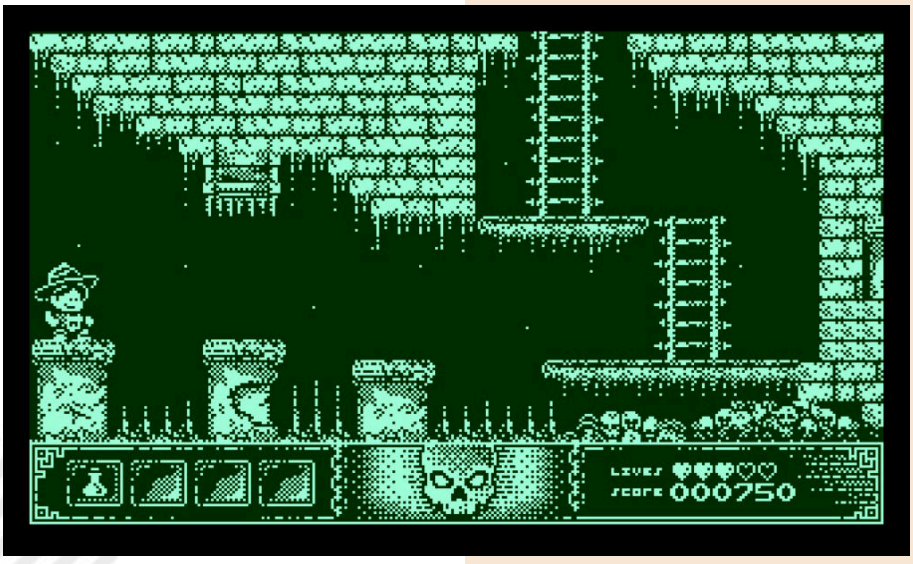


Fig. 7,8 - Immagini tratte dalla versione Atari 8 bit. Ancora una volta monocromatico, ma questa volta in una tonalitá di verde.



Aggiornamento Ottobre 2024.
Gli ultimi aggiornamenti del gioco migliorano fluidità e gameplay e regalano (per gli utenti Atari8bit) il simpatico Go!Tony Go!, un running game su carrello minerario. Un simpatico giochino che aggiunge qualche punto al voto finale

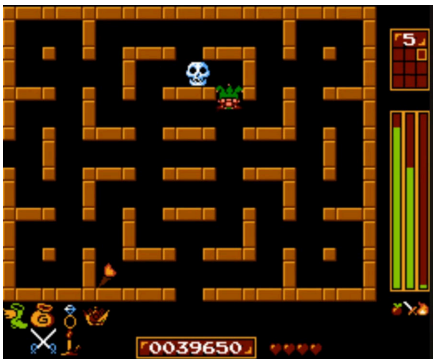
di **Carlo Nithaiah Del Mar Pirazzini**

NEW GAME

JESTER

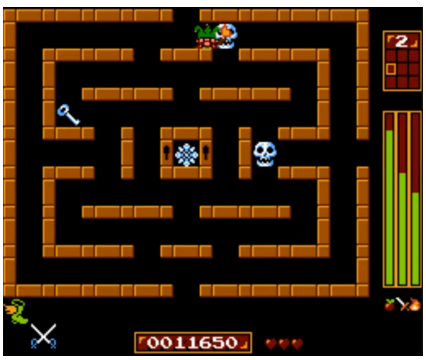
Anno: 2024
Editore/Sviluppatore: MegaCat Studio
Genere: Maze game
Piattaforma: NES/Super Nintendo
Sito web: https://www.collectorvision.com/games/jester.php

Jester, il buffone del Re. Fai sempre ridere tutti ma questa volta c'è poco da ridere.
Chiuso in un dungeon, inseguito da mostri terribili e con lo scopo di raccogliere tutti gli ogetti Elementali perduti.
Difficile vero! Beh, la vita è uno scherzo!
Simpatico questo Jester, un titolo uscito nel 2023 e in versione "fisica" nel 2024 per Nintendo 8 bit e Super Nintendo.
Una lettera d'amore a Le Fou du Roi, uno dei giochi disponibili sul sistema TV via cavo Videoway negli anni 90. Un sistema di tv interattivo analogico disponibile solo in Canada tra il 1990 e il 2006. Evidentemente i MegaCat lo hanno amato molto per riproporlo adesso.
E' descritto come "un classico gioco di inseguimento di labirinti" e ricorda le meccaniche d'azione dei tanti titoli simili.
Interessanti gli oggetti da raccogliere che permettono al nostro "buffone" di combattere i mostri, di continuare a vedere la luce all'interno dei dungeon e di ricaricare l'energia.



Le versioni Nes e Super Nes sono identiche, non cambia nemmeno la velocità di gioco.
Come dicevo il gioco è disponibile in versione fisica con un bel artwork di copertina e un manuale curato e molto colorato.
Jester però soffre di noia. Il gioco è davvero tutto uguale e dopo un po' ci porterà a spegnere la console.
Peccato perché il protagonista è davvero buffo.

di **Marta Rossmann**



GIUDIZIO FINALE

» Giocabilità 70%
Un titolo semplice da giocare.

» Longevità 60%
Dopo i primi livelli la noia ci as6sale in modo più aggressivo dei vari mostri presenti.

NEW GAME

DR. DANGEROUS
SECRETS OF THE TEMPLE OF XOL'TAN

Anno: 2024
Editore/Sviluppatore: HooGames2017
Genere: Platform/Puzzle
Piattaforma: Amiga
Sito web: https://hoogames2017.itch.io/dr-dangerous

Il dottor Victor Dangerous è un tipo strambo. Ha scoperto un tempio segrete nella foresta amazonica, ci ha costruito un suo laboratorio per esperimenti dentro e gira tutto il giorno in tuta Hazmat. Con quel caldo!? Folle!

Ora, una sera durante una tormenta, un fulmine colpisce l'alimentazione elettrica del laboratorio. Tutto si guasta e si attivano i protocolli di sicurezza automatici. Le porte si chiudono e il povero dottore viene violentemente gettato in un profondo tunnel sotterraneo.

Un tunnel pieno di pericoli che vedrà il nostro eroe affrontare le creature più pericolose e bislacche e recuperare le batterie di ripristino per riattivare le funzioni del tempio. Ovviamente non sarà impresa facile, le creature posseggo un alto tasso di aggressività innata e il nostro dottore è armato di una pistola che si scarica subito.

Insomma, avete capito bene siamo di fronte al classico platform che ci mette di fronte piccoli enigmi per superare i vari livelli e che non ci rende le cose semplice. La pistola va ricaricata recuperando le cariche di fuoco e sarà l'ago della bilancia per superare moltissimi ostacoli tra noi e l'agognata libertà.

Dr. Dangerous è un gioco che non inventa nulla. Ricorda Rick Dangerous e anche il recentissimo Tony Montezuma. Livelli ben fatti e profondi pieni di trappole e mostri che vanno completati seguendo logica e abilità manuale.

Questa sua mancanza di innovazione

però non è un limite e nemmeno un problema. Dr Dangerous è ben fatto e si gioca con gusto, ennesima dimostrazione che se si curano aspetto tecnico, gameplay e level design si otterrà un buon prodotto anche se l'innovazione è zero.

Mi è piaciuto esplorare i livelli

GIUDIZIO FINALE

» Giocabilità 80%
Nulla di nuovo ma quello che c'è è ben fatto e il gioco è un buon esempio di equilibrio, level design e gameplay.

» Longevità 85%
La curva della difficoltà è ben bilanciata e il sistema di codici per salvare i livelli è semplice e funzionale.

sotterranei del tempio e il grado di sfida cresce in modo graduale senza mettere mai in difficoltà il giocatore.
Interessante il sistema di codici per ripartire dal livello dove si è "morti" e la possibilità di giocare con il secondo tasto per il salto.
Insomma, non sarà il platform della vita su Amiga ma è un buon prodotto realizzato con il sempre più efficiente Scorpion Engine (e si vocifera di una versione per Megadrive).
Il titolo gira su tutti gli Amiga 500 PAL con estensione da 512Kb e kickstarter 1.3 e ovviamente in emulazione e su A500 mini.

di **Carlo Nithaiah Del Mar Pirazzini**

NEW GAME

PARASOL STARS
THE STORY OF BUBBLE BOBBLE III

Anno: 2024
Editore/Sviluppatore: ININ Games, Taito
Genere: Platform
Piattaforma: Nintendo Switch, Playstation 5, Xbox Serie X/S, Playstation 4, Xbox One
Versione di prova: Nintendo Switch

Sembra quasi che Parasol Stars sia il capitolo "dimenticato" della saga di Bubble Bobble.

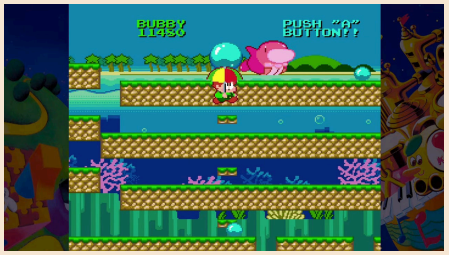
Certo, ha ben poco a che fare con il gioco originale del compianto Maestro Fukio Mitsuji per quanto riguarda aspetto e gameplay. Ma il tono è allo stesso livello, e accidenti se è così spassoso da giocare oggi come nel 1991, anno in cui fu lanciato su Turbo Grafx16.

Ora ININ Games ha riportato il titolo alla sua piena gloria sulle console moderne e noi di Retromagazine World ce lo siamo giocati su una moderna Switch. Lo ha fatto ad un prezzo esiguo, solo dieci euro e ad un sacco di opzioni. Ma, cosa più importante, è ancora divertente dopo tutti questi anni soprattutto se giocato in doppio.

Personalmente avrei visto una compilation/trilogia di Bubble Bobble, però è bello vedere Parasol Stars ricevere l'attenzione meritata.

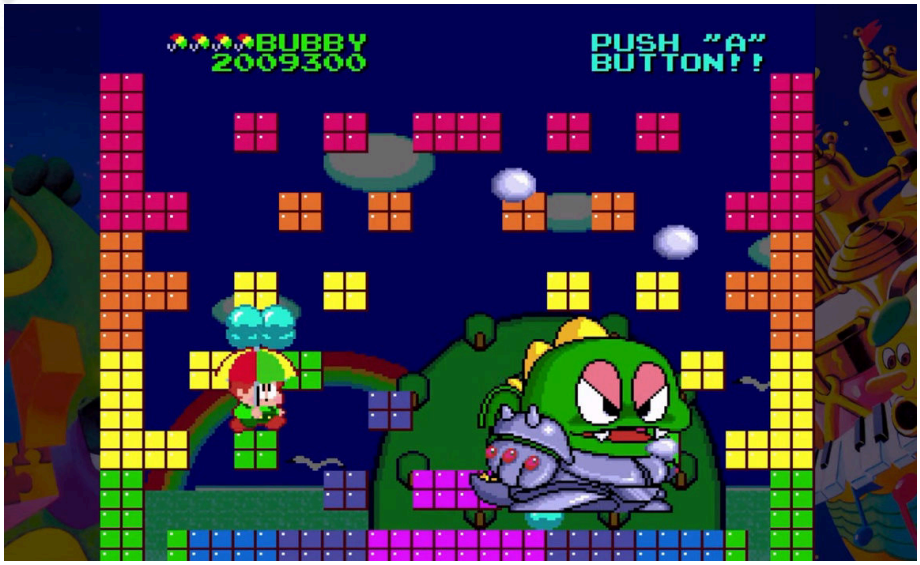
Sebbene il gameplay sia diverso rispetto ai precedenti titoli della saga (Bubble Bobble e Rainbow Islands), Parasol Stars ha principi simili. Si deve liberare uno schermo zeppo di nemici, usando un ombrellino e diversi modi creativi. Questi includono raccogliere i nemici sulla punta del parasole e lanciarli contro altri nemici, così come raccogliere bolle per effettuare terribili e roboanti attacchi speciali.

Ci sono alcune strategie più profonde come il concatenare diversi nemici colpiti o raccogliere più frutta per aumentare punteggio, vite e bonus. Poi, alla fine di ogni serie di livelli, si affronta un boss più o meno impegnativo. Per inciso le boss fight sono ancora molto divertenti da affrontare.



ININ Games ha fatto un lavoro magistrale nel portare questo classico di Taito al suo splendore nell'era moderna. Ho particolarmente amato le versioni Amiga e ST del gioco e devo ammettere che vederlo su Switch in modalità portatile mi ha rimandato all'estate del 1992.

I ragazzi del team di sviluppo hanno anche aggiunto diverse opzioni interessanti: i riavvolgimenti in game,



GIUDIZIO FINALE

» Giocabilità 90%
Un titolo divertente anche dopo trent'anni.

» Longevità 90%
La curva di difficoltà graduale lo fa apprezzare ancora di più.

un sistema di save state/load state moderno, numerose opzioni di gioco e tanto altro ancora. E, ovviamente, la bellissima versione in doppio da giocare in locale sia in modo cooperativo che competitivo.

Nel suo complesso, tutt'ora Parasol Stars è un gioco meraviglioso che vale la pena riscoprire se non lo giocate da un po' ed è l'ideale punto di congiunzione tra noi e i nostri figli/nipoti. Un modo per mostrare come la semplicità di design e il bel gioco siano divertenti e travalichino l'età differenti. Ma come ci appare con gli occhi di trent'anni dopo? La grafica regge bene dopo tutto questo tempo, e il design degli scenari è luminoso, colorato e ben riconoscibile. È bello rivedere alcuni "vecchi" nemici tra un livello e l'altro. Sanno di ricordi e di the alla pesca nei lontani pomeriggi d'estate della nostra memoria.

La musichetta è sempre orecchiabile e mai fastidiosa. Un motivetto allegro sulla falsariga di Bubble Bobble.

Regge il confronto con l'era moderna? Si, e piuttosto bene. Non pretende di essere innovativo o un must assoluto ma resta un gran bel gioco di piattaforme, magari da giocare durante una giornata uggiosa o in treno in modalità portatile.

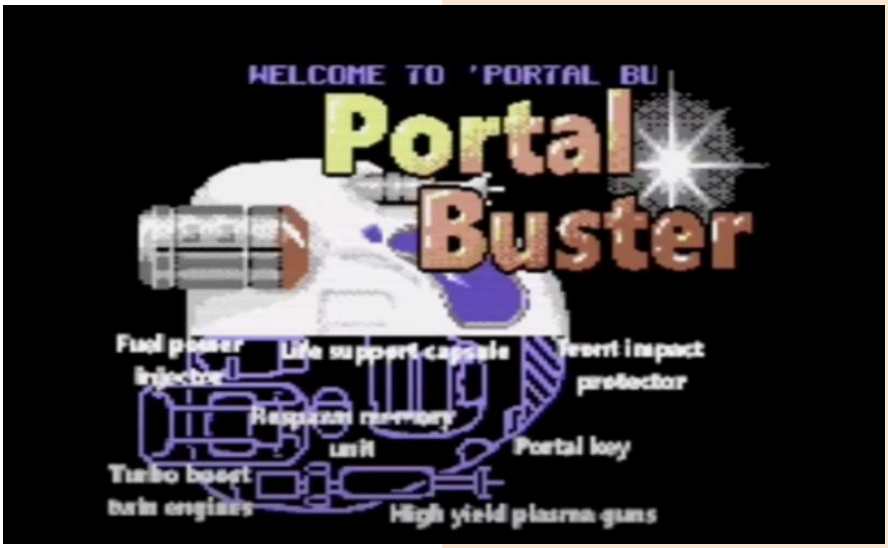
di **Carlo Nithaiah Del Mar Pirazzini**

NEW GAME

PORTAL BUSTER

Anno: 2024
Editore/Sviluppatore: drmortalwombat
Genere: Shoot em up
Piattaforma: Commodore 64
Sito web: https://drmortalwombat.itch.io/portal-buster

Veloce è la parola d'ordine! Anzi Super Veloce! Questo sicuramente è il primo pensiero che vi viene in mente giocando a Portal Buster, nuovo shooter per il nostro caro C64.



Un titolo decisamente inaspettato che abbiamo particolarmente gradito e che ricorda il classico Uridium ma anche Armalyte e che, soprattutto per la sua già citata velocità, mette a dura prova anche i giocatori più abili di questo genere.

Lo scopo del gioco è quello di distruggere un mega boss nemico, raggiungendolo attraverso una mappa vastissima e attraversando innumerevoli ostacoli e nemici.

Il gioco è solo PAL e usa VSP per lo scorrimento ma è pienamente compatibile con C64/128, emulatori vari e TheC64 family.

E' divertente e di questi tempi è già un plus! Graficamente è gradevole e

Se questo non bastasse, l'astronave ha anche gravi problemi di carburante e deve rifornirsi raccogliendo dei barili lungo lo schermo.

Lo stile a multi-scorrimento come in Uridium permette di regolare la velocità e di rallentare il ritmo di gioco.

Lo sviluppatore ha anche aggiunto un paio di "trainer" per rendere meno letale l'esperienza di gioco e per questo lo ringraziamo infinitamente. Effettivamente 15 secondi tra un rifornimento e l'altro sono decisamente pochi!

GIUDIZIO FINALE

» Giocabilità 80%
Un clone di Uridium dalla velocità pazzesca e dal buon sistema di gioco. I trainer aiutano a prendere la mano con lo stile di gioco.

» Longevità 75%
Ostico e non per tutti, ma davvero gradevole una volta appreso al meglio.

ha un buon comparto audio grazie al sapiente uso del SID.
Forse è un po' ostico alle prime armi ma vi assicuriamo che non vi staccherete tanto presto dal joystick e una partita terminata male ne chiamerà subito un'altra.

Consigliatissimo per gli appassionati del genere e per gli amanti dei titoli "harder, faster, stronger".

di **Carlo Nithaiah Del Mar Pirazzini**

NEW GAME

BINARY BATTLE

Anno: 2024
Editore/Sviluppatore: Haplo
Genere: Dungeon Crawler
Piattaforma: Commodore 64, Plus/4, ZX Spectrum
Sito web: https://h4plo.itch.io/binary-battle

Ti sei appena svegliato e piazzato davanti al tuo fidato Commodore 64. Hai afferrato il tuo floppy preferito ma ti accorgi che è pieno zeppo di virus! Maledizione! L'unica copia esistente del file sorgente del gioco a cui hai lavorato negli ultimi sei mesi... Niente paura! Il tuo potente antivirus ti aiuterà nel ripulire tutto e ricominciare a lavorare.

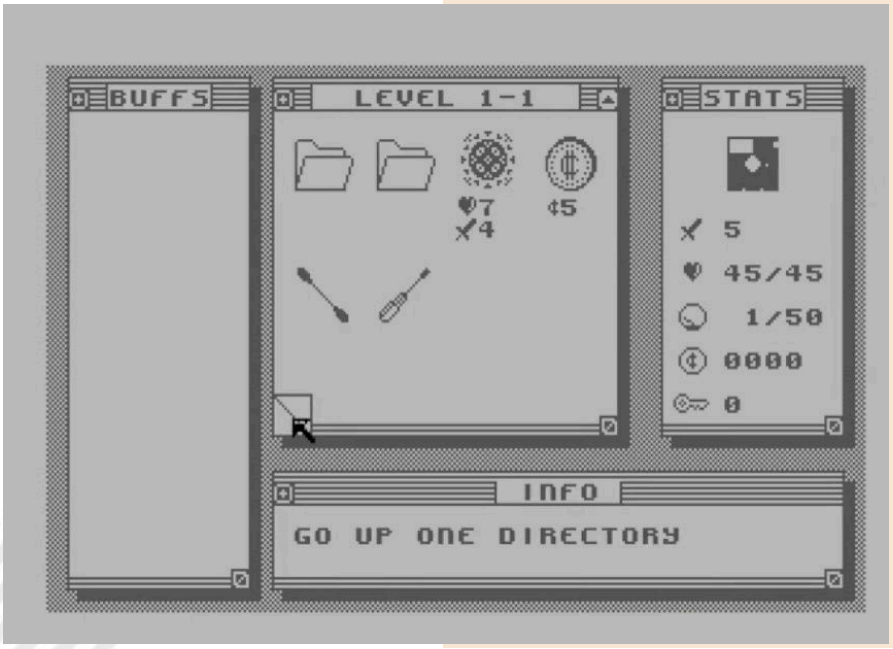
Binary Battle è una nuova interpretazione del genere dungeon crawler ed è geniale.

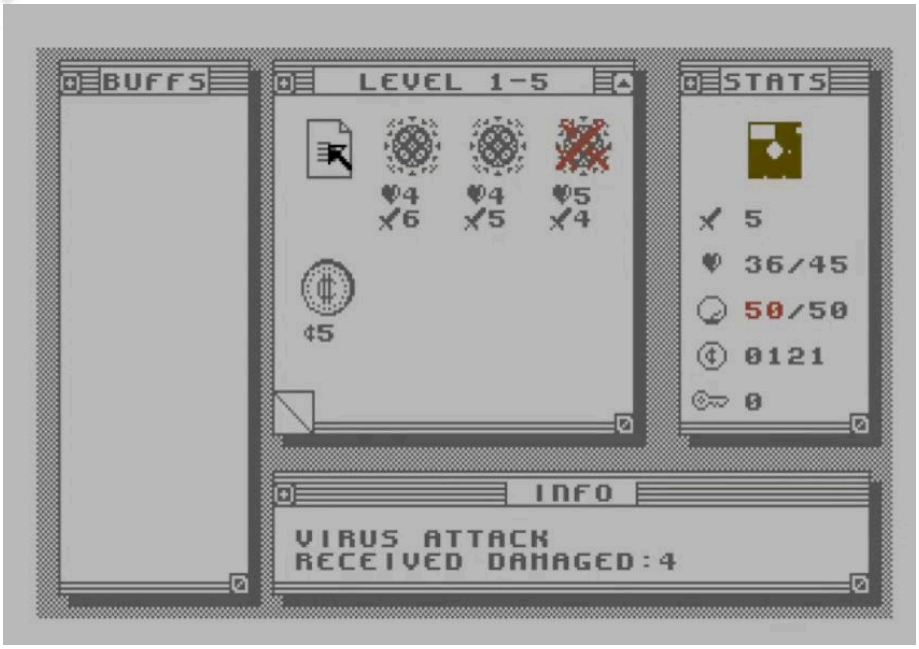
Ispirato al GEOS e al gioco su PC Folder Dungeon, ci pone come obiettivo quello di recuperare il codice attraversando cinque ostici livelli di gioco pieni di errori, virus e dati danneggiati.

Con il cursore del mouse ci muoveremo nell'ambiente GUI per colpire i file corrotti o le cartelle e ripulirli. All'occorrenza potremo riparare i danni subiti, recuperare monete virtuali (utili per il potenziamento) e ripristinare il sistema.

Il nostro antivirus ha una salute limitata che diminuisce ad ogni attacco dei virus. Inoltre, ogni azione genera riscaldamento e la nostra floppy unità potrà subire dei danni.

Un titolo davvero curioso che ci ha conquistati. L'apparenza inganna. Ciò che inizialmente sembra solo una bella rappresentazione grafica di un desktop GUI Geos Style finisce per rivelarsi un roguelike immersivo con diversi elementi strategici coinvolgenti. Gestire la salute del proprio disco mentre si eleminano virus e si naviga tra le directory e le strutture dei file è un'esperienza coinvolgente, con fallimenti che spesso si traducono in





GIUDIZIO FINALE

» Giocabilità 90%

Buon livello di strategia unito ad uno stile unico e accattivante. Aggiungiamoci poi le partite random ogni volta che ricominciamo e una gradevolissima soddisfazione ogni volta che "ripuliamo" il nostro programma.

» Longevità 80%

Lo stile potrebbe scoraggiare il giocatore casuale e forse 5 livelli sono troppo pochi.

un altro tentativo per vedere se si riesce ad arrivare all'obiettivo.

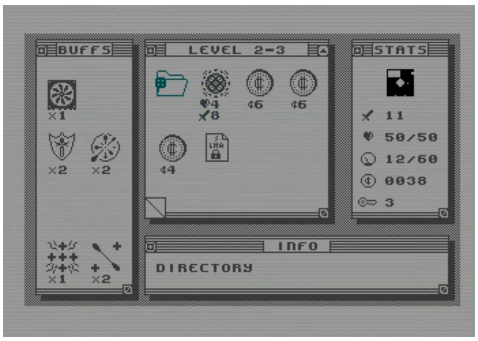
Lasciate perdere la grafica elaborata, qui abbiamo un vero piccolo capolavoro di gameplay (e una gran bella colonna sonora).

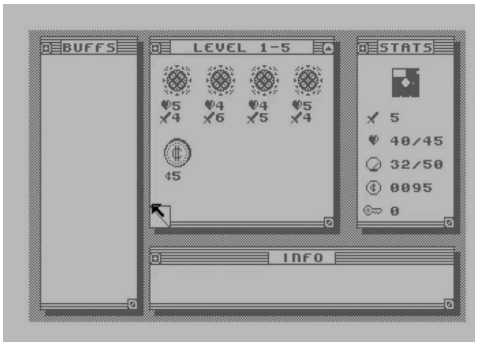
Le tre versioni disponibili differenziano minimamente l'una dall'altra. La struttura del gioco rimane la medesima e, se cercate un titolo da apprendere e che vi coinvolga, Binary Battle è un'esperienza unica e gratificante.

di **Carlo Nithaiah Del Mar Pirazzini**

Versione Plus 4 (sopra)

ZX Spectrum (sotto)





NEW GAME

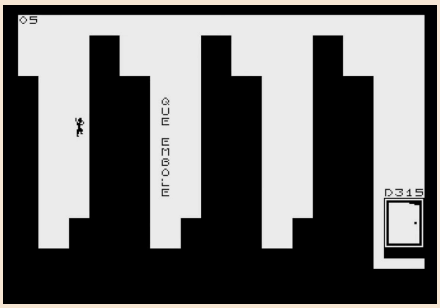
SHIFT

Anno: 2024
Editore/Sviluppatore: Haplo
Genere: Puzzle/Platform
Piattaforma: Commodore 64, AtariST, Amiga, Atari 8bit
Sito web: https://h4plo.itch.io/shift

Shift giunge a noi nel lontano 2008, in un gioco in Flash realizzato da Antony Lavelle e Armor Games e diventa in poco tempo un successo virale. In poco tempo l'autore pubblica ben 3 seguiti e numerosi porting per svariate piattaforme portatili e non. Immediato, avvincente, semplice da giocare e da guardare. Immancabile per tutti i videogiocatori alternativi. Questo demake per retro piattaforme è bellissimo. In ogni sua versione.

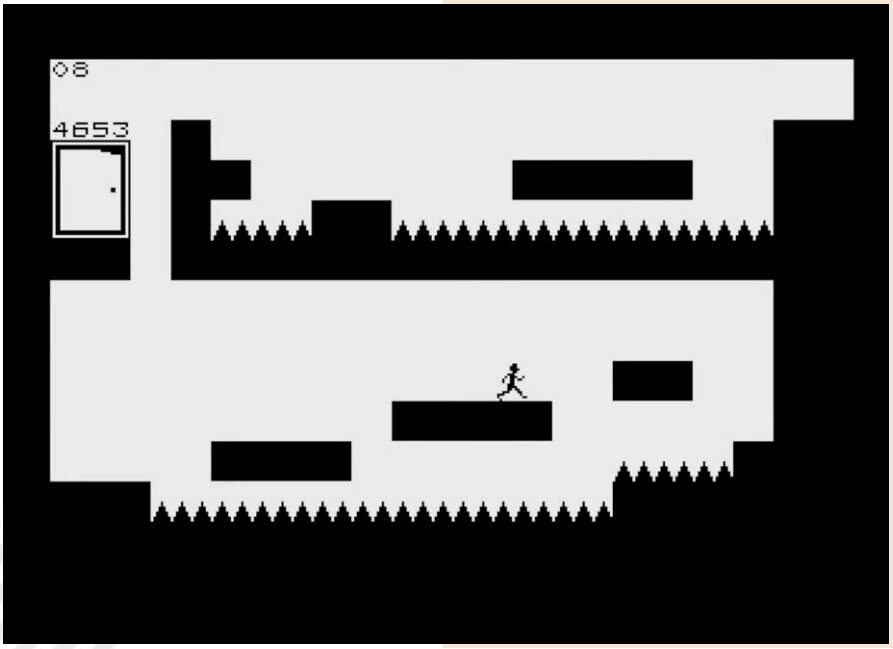
Il gioco sembra semplice ma ci porta a ragionare su come portare fuori da ogni livello il nostro stilizzato protagonista tutto nero. Tra salti impossibili e massima precisione il bello è nello "shifttare" tra nero e bianco capovolgendo lo stile di gioco per poter "rendere" più semplice il percorso.

Minimalismo assoluto per un divertimento totale! Moltiplicazione della giocabilità.

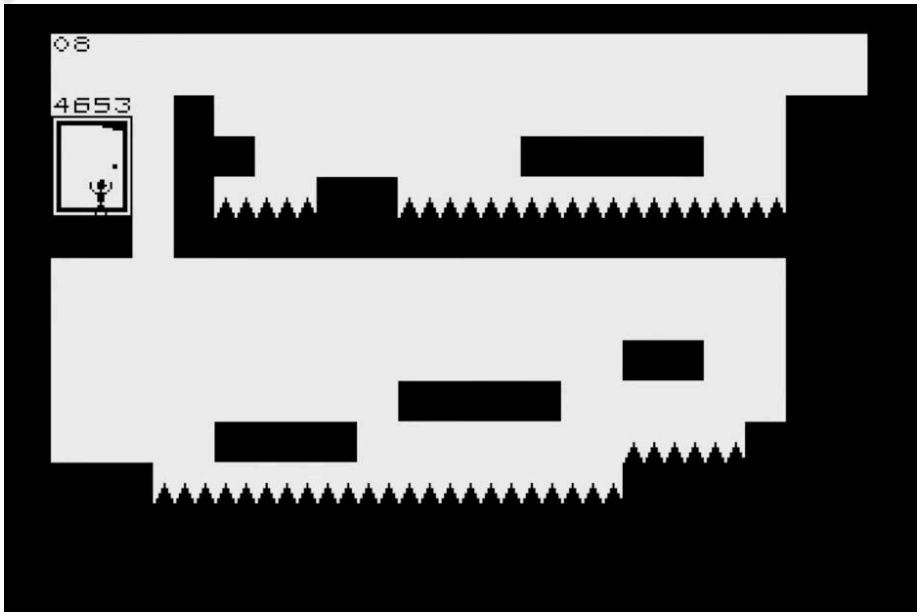


Le quattro versioni si equivalgono. Più dettagliato su Amiga e C64, ma decisamente bello da vedere e giocare anche sugli 8bit Atari.

Questo funzionava in FLASH e funziona perfettamente su Commodore 64, Atari8bit, ST o Amiga. Stessa giocabilità e stessa voglia di mettersi in gioco per capire come approcciare i numerosi quadri.



Tutto funziona perfettamente con una grafica minimalista e bellissima allo stesso tempo su tutte e quattro le piattaforme, un suono pregevolissimo (su Amiga davvero bello!) e il già solido gameplay citato.



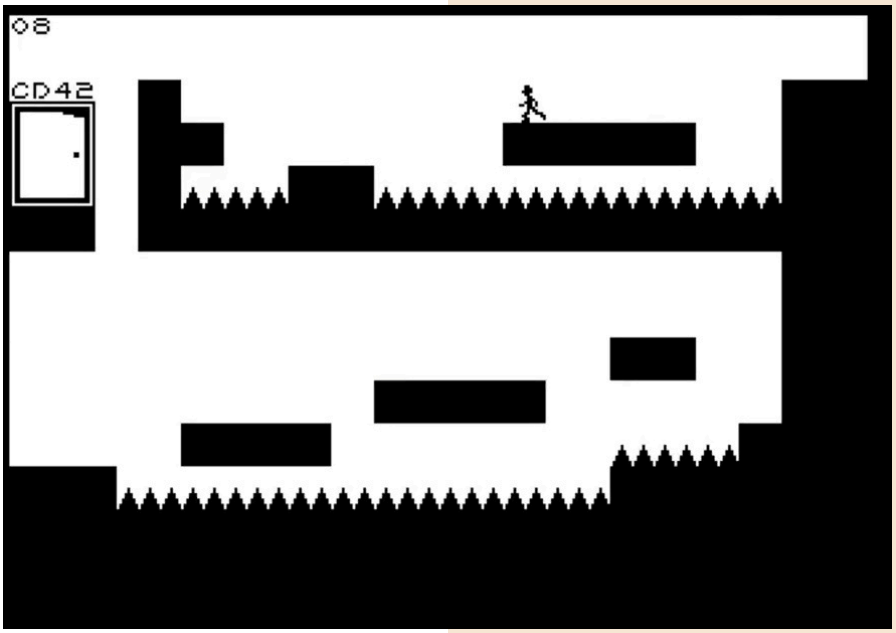
GIUDIZIO FINALE

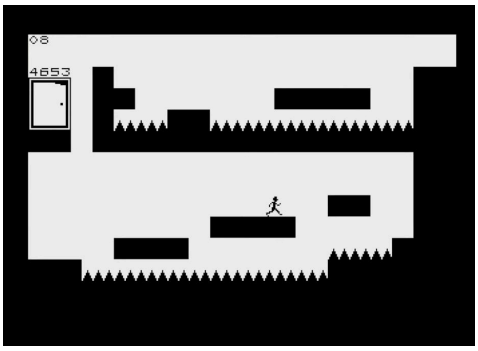
» Giocabilità 90%
Meccanica di gioco splendida.

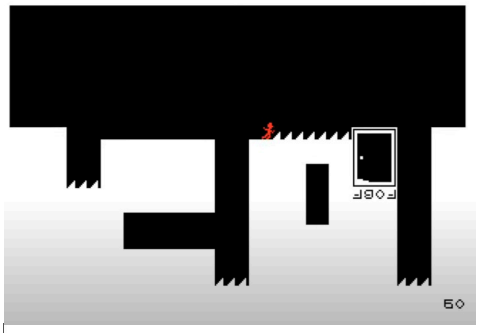
» Longevità 80%
Il level design impegnativo ma mai frustrante vi permetterà di apprezzarlo partita dopo partita e i suoi 37 livelli (44 su Amiga) vi occuperanno molte molte ore.

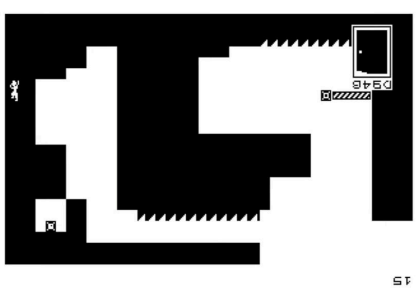
Un altro centro di Haplo che sicuramente continuerà a lavorare su un possibile porting per altre retropiattaforme e su una (difficilissima) conversione del secondo capitolo.

di **Marta Rossmann**



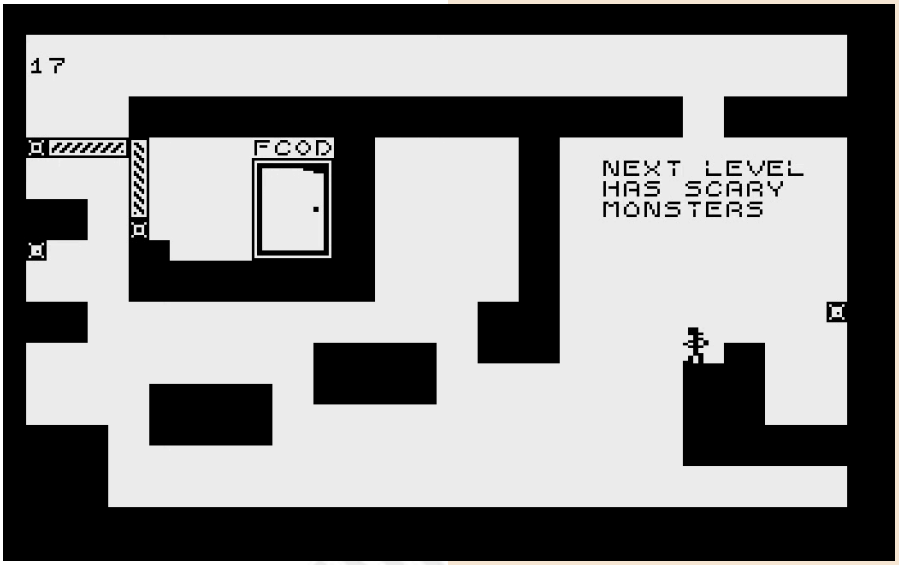






Versione Amiga (sopra)

Atari 8bit (sotto)



METAL SLADER GLORY DIRECTOR'S CUT

Anno: 2000
Editore/Sviluppatore: HAL Laboratory
Genere: Avventura grafica/ testuale
Piattaforma: Super Famicom

Nel 2000 eravamo già in piena era PS2 ma in Giappone il Super Famicom (il nome giapponese dello SNES) continuava ad essere molto apprezzato. Metal Slader Glory Director's Cut è stato l'ultimo titolo uscito su questa console e fu pubblicato il 29 novembre del 2000.

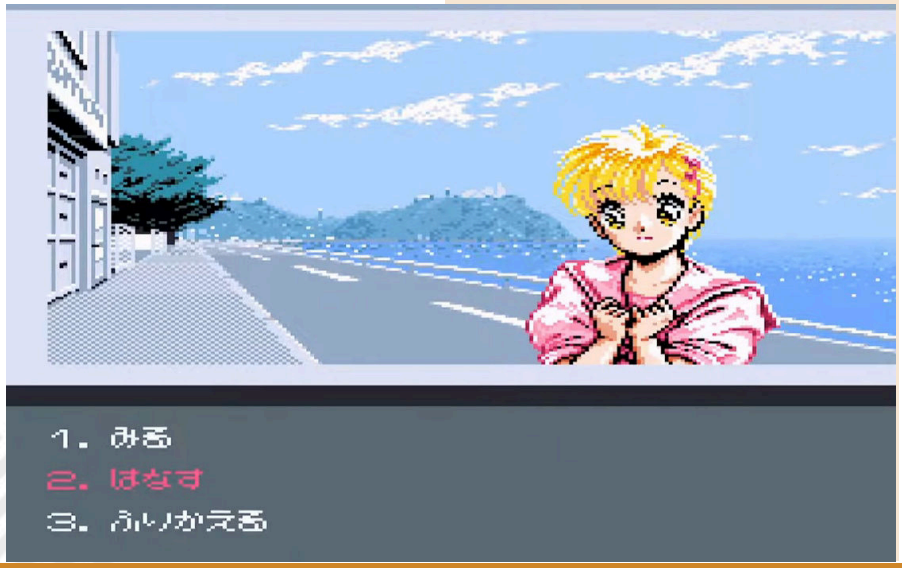
Mentre in Europa la pubblicazione di titoli si fermò nel 1997, nel paese del Sol Levante continuò fino a tutto il 2000, dimostrando una fedeltà non comune per la macchina a 16bit di Nintendo da parte di uno zoccolo duro di appassionati.

Questa Director's Cut è la versione migliorata del titolo originario del 1991 uscito su NES. Si tratta di una visual novel realizzata dal mangaka Yoshimiru Hoshi.

Per quanto sia un genere quasi sconosciuto in occidente, presenta un gameplay perfettamente bilanciato e una storia avvincente. La grafica incredibile completa perfettamente il gioco, facendolo sembrare di gran lunga più avanti ad altri titoli simili sul mercato. Il suo unico difetto? La mancata traduzione ufficiale in lingua inglese.

Metal Slader Glory si svolge come una normale visual novel, ma con divertenti deviazioni dal gameplay standard. L'esperienza di gioco normalmente ci porterà a chiedere in giro indizi per raggiungere la posizione successiva. Come risultato dell'incredibile quantità di testo, si possono attivare molti più dialoghi semplicemente usando la stessa opzione due volte. Tuttavia, si sperimenteranno altri stili di gameplay, come una missione dungeon crawler o una ispirata a un gioco di strategia. Atipico per il genere ma decisamente divertente.

Come visual novel, la storia è ovviamente l'aspetto più forte del



gioco. Comincia il tutto con Tadashi, il nostro protagonista, e la sua amica Elina che provano un mech da lavoro che hanno comprato da Gen's Auto. Quando Tadashi ci salta dentro, la copertura si stacca e si rivela essere uno Slader, mech da guerra usati durante un conflitto nello spazio, che sono stati presumibilmente smantellati dopo la guerra. Tuttavia, questo Slader invia un messaggio minaccioso che dice "Terra in pericolo. Cerca subito il creatore". Incuriositi da questo, lui, Elina e la sorella di Tadashi, Azusa, si dirigono a casa di Gen per saperne di più. Questa semplice premessa porta a un conflitto più grande di quanto chiunque di loro possa immaginare. La storia inizia abbastanza bene e stabilisce la trama in modo organico, dandoci al contempo la personalità dei personaggi. L'aspetto visivo è decisamente ben fatto. Colori vivaci, personaggi ben caratterizzati e sfondi stupendi. La parte musicale è il pezzo forte, viva e vibrante di emozioni.

In rete esistono alcune patch di traduzione per rendere più comprensibile i testi e la storia, ma con un po' di pratica è possibile portarlo a termine anche senza conoscere il giapponese, ovviamente perdendo la narrativa e soprattutto i bellissimi dialoghi finali. Ricorda le avventure testuali del passato ma con una trama mecha tipica della fine degli anni 90/inizio 2000. Un titolo storico che vi invito a riscoprire se amate il genere, i manga e le tematiche mecha apocalittiche.

di **Roberto Del Mar Pirazzini**

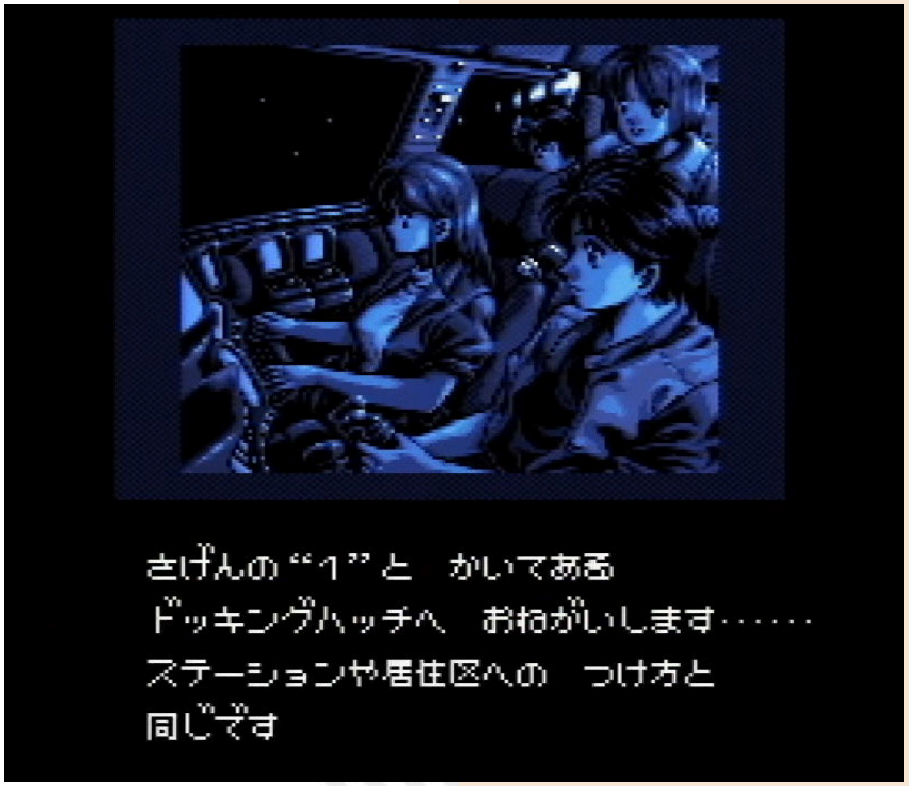
GIUDIZIO FINALE

» Giocabilità 80%

Non è difficile apprendere le meccaniche di gioco nonostante la lingua. Alcune innovative sezioni lo rendono meno ripetitivo delle visual novel del tempo.

» Longevità 80%

La storia, se compresa, vi terrà attaccati al pad. Un prodotto per veri OTAKU.



NEW GAME

QYXAR

Anno: 2024
Editore/Sviluppatore: 8bit Milli Games/William Hicks
Genere: Shoot em up
Piattaforma: ColecoVIsion
Sito web: https://consolemoo.com/shop/ols/products/rom-qyxar-colecovision-rom-dng-pln-clc1

Sono più di quarant'anni che alieni, mostri, creature meccaniche e malvagi pazzoidi cercano di distruggere, conquistare, annientare la razza umana.

Questo, ovviamente, accade nei videogiochi dove prodigiosi eroi nelle loro navi spaziali fronteggiano e distruggono il pericolo.

Eppure, dopo tutto questo tempo continuano ad invadere e questa volta lo fanno attaccando i nostri ColecoVision.

Qyxar è uno shoot em up a schermata fissa che ci vede impegnati a ripulire la galassia da una razza aliena insettoide piuttosto brutale.

Nel più classico di questo genere di giochi ci troveremo ad affrontare orde su orde si mostruosità ronzanti bellicose. Non sarà impresa facile. Le wave d'assalto diventano, con il proseguire dei livelli, sempre più aggressive e veloci.

Dalla nostra avremo uno scudo di resistenza che ci protegge dai colpi (ovviamente degradandosi) e dei power up essenziali che aumenteranno le nostre capacità difensive.

Interessante la possibilità di muoversi sullo schermo e di non rimanere statici, caratteristica che mette un po' di pepe in più al titolo.

Qyxar si è rivelato un buon titolo, divertente e sviluppato con cura disponibile in edizione digitale a soli 14 dollari.

Personalmente avrei creato diversi tipi di creature e lavorato su un maggior numero di pattern per le wave avversarie, ma è una semplice opinione personale.

Promosso!

di **Carlo Nithaiah Del Mar Pirazzini**

GIUDIZIO FINALE

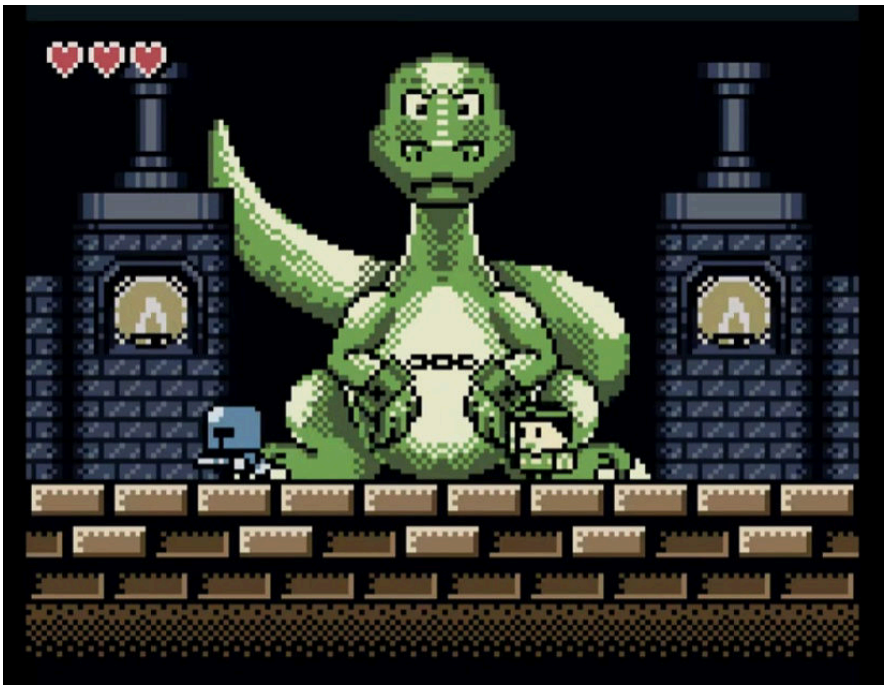
» Giocabilità 80%
Un prodotto ben sviluppato con alcune interessanti caratteristiche.

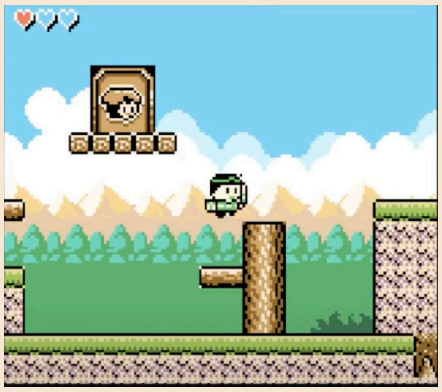
» Longevità 70%
Titolo tosto progredendo nei livelli. Alla lunga potrebbe stancare per la ripetitività dell'azione e le wave piuttosto simili tra loro.

NEW GAME

POKETTOHIRO!

Anno: 2024
Editore/Sviluppatore: Broke Studio!/David Marin
Genere: Action/Platform
Piattaforma: Game Boy Color
Sito web: https://pokettohiro.com/





Eroe portatile! Eroe in stile kawaii-super deformed e non solo! Abbiamo di fronte un gioco davvero coinvolgente che mi ha ricordato alcuni titoli del passato come Wonder Boy in Monster Land (il gioco preferito di mio padre). È una bella avventura dove un gruppo di eroi coraggiosi dovrà raccogliere i frammenti del Cristallo Magico per impedire ad un malvagio Cavaliere Nero di distruggere il regno.

Ha un gameplay "aperto" proprio come Monster Land e la possibilità di scegliere differenti eroi (ben 6) dalle caratteristiche di gioco differenti. Dallo spadaccino, all'arciere, dal guerriero falco alla fata. Tantissima varietà in un titolo che omaggia i classici del passato.

La storia sembra banale ma coinvolge molto e il gioco promette una po' di ore di divertimenti prima di essere concluso (una ventina basteranno!). I personaggi giocabili hanno ognuno la propria fisica e le proprie abilità. Queste ultime necessarie per superare ostacoli, sconfiggere nemici e scoprire i diversi segreti nascosti dentro al gioco.

Mi è piaciuto esplorare il micro mondo di Pokettohiro!, mi ha coinvolta tanto e mi ha portato a visionare ogni singolo fotogramma della mappa.

È un titolo nato in digitale acquistabile sulla pagina itch.io degli sviluppatori, ma che rende tantissimo in versione fisica. La cartuccia è bella da vedere. Un bel packaging con un ciccioso manuale di 32 pagine e alcuni gadgets.

Concludo parlando dell'aspetto esteriore. Musica semplice ma coinvolgente e una gran bella grafica dettagliata e ricca di particolari.

Se cercate un buon prodotto per GBC questo è proprio il titolo che fa per voi.

di **Ingrid Poggiali**

GIUDIZIO FINALE

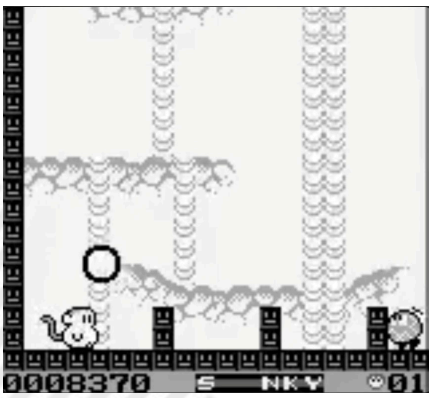
» Giocabilità 92%
Tanti personaggi giocabili tutti differenti e una bella storia avvincente. Semplice nel gameplay e con delle sfide mai banali.

» Longevità 90%
Il gioco va esplorato tutto e va vissuto con lo stesso spirito dei grandi classici del passato. Bello!

SPANKY'S QUEST

Anno: 1991
Editore/Sviluppatore: Natsume, Taito
Genere: Platform/Action
Piattaforma: Game Boy

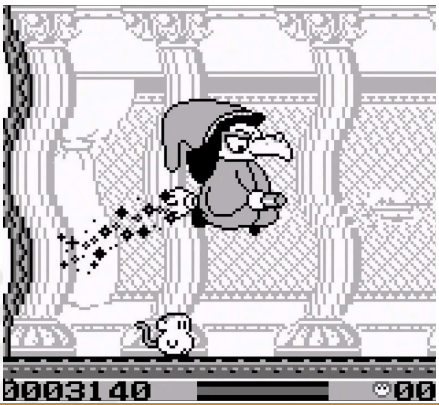
Quanto è bello Spanky's di cui ero già a conoscenza ma che non avevo ancora giocato!
Un titolo sulla scia di "Bubble Bobble" e "Snow Bros." con la differenza che è un gioco action, anche se non sembra.
Il nostro protagonista è una scimmietta paffuta che ha come arma una piccola bolla che fa rimbalzare sulla testa e che possiamo far esplodere a nostro piacimento con varie misure. Piccola fornirà un colpo e quella grande esploderà in tante bolle uccidendo così i nemici... durante il gioco possiamo raccogliere anche le varie lettere che completeranno la scritta "Spanky" che eliminerà tutti i nemici (che sono dei frutti) e ci porterà al bonus stage dove facendo rimbalzare una bolla enorme possiamo accumulare punti.

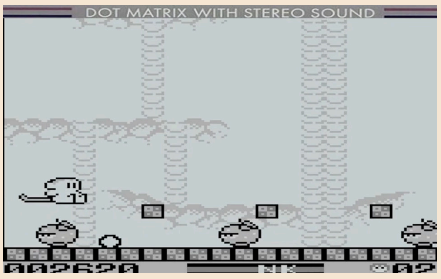


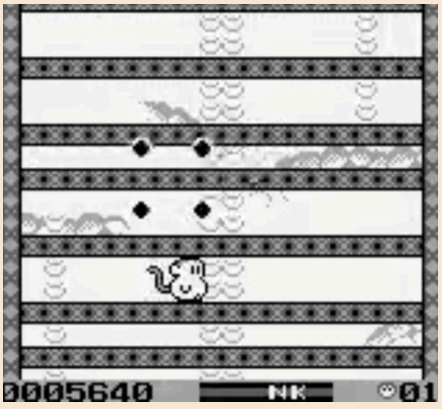
Il gioco è composto da 5 stage: 1-16, 2-16, 3-16, 4-16, 5-05, ogni livello ha un Boss finale mentre nel 5° troviamo tutti i Boss (anche loro sono dei frutti ma giganti) fa eccezione solo la Strega antagonista di tutta l'avventura che una volta sconfitta ci mostra il finale.

Questa versione è molto simile alla controparte Nes (con le dovute differenze), ed è ambientata in una torre dove il nostro paffuto protagonista è prigioniero della Strega. Bella la grafica e un buon accompagnamento sonoro, il gioco è un bel attimo di svago e ci si passano volentieri le ore libere (poche) che abbiamo.
Recuperatelo e giocateci in qualsiasi forma possibile, anche se in cartuccia e su Gb fa il suo degno figurone.
E noi comunque ci vediamo al prossimo post!
Baci, Miss Game Boy

di **Barbara Murgida**







GIUDIZIO FINALE

» Giocabilità 85%
Un classico platform con molta azione e tanto da giocare. Semplice e divertente.

» Longevità 80%
Non impossibile da finire ma sicuramente coriaceo. La strega mi fa impazzire!

NEW GAME

THE CURSE OF LIES

Anno: 2024
Editore/Sviluppatore: Rei-Vax-73
Genere: Puzzle Game
Piattaforma: MSX2, MSXDOS2
Sito web: https://www.msxdev.org/2024/10/09/msxdev24-11-the-curse-of-lies/

Lo sviluppatore francese Rei-Vax-73 ci presenta questo puzzle game interessante in concorso al MSXdev24. La competition è una delle più seguite e agguerrite "gare" di sviluppo per lo standard MSX e presenta ogni anno dei piccoli gioielli.

The Curse of Lies ci vede impegnati ad assemblare la corretta combinazione di fiori mistici per scoprire il Fiore della Verità, che aiuterà due stregoni, un lui ed una lei, a togliersi una maledizione terribile. Da poveri apprendisti il compito non sarà facile. Le combinazioni ci verranno suggerite dai due mistici in alto nello schermo e noi dovremmo trovare il giusto colore del fiore per poter proseguire nei livelli. Ad ogni errore si perderà una vita fino al temuto game over.

Facile vero? Insomma, le combinazioni cromatiche possono trarre in inganno e per questo il gioco è scaricabile con tanto di manualetto ed istruzioni per poter essere compreso al meglio.

Un gioco molto intrigante per meccanica che guadagna davvero molto in termini di giocabilità e longevità se approcciato in doppio.

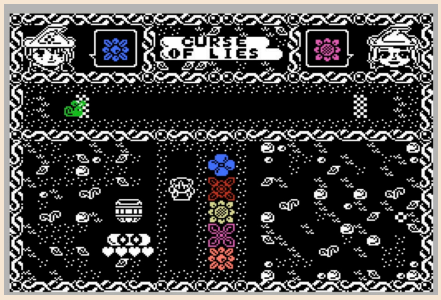
Graficamente è ben fatto e minimalista e il comparto sonoro è gradevole e non invasivo.

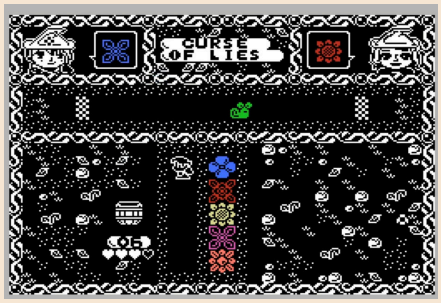
La meccanica delle combinazioni può spiazzare senza aver letto bene il manuale ma, appresa una volta, difficilmente sbaglierete in gioco.

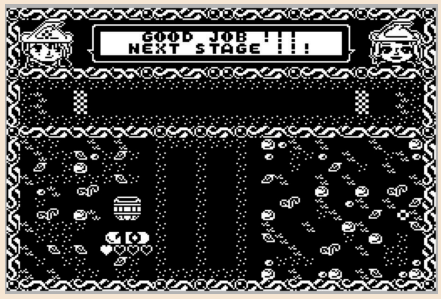
Se giocato in singolo soffre molto di ripetitività mentre in doppio, a causa della competizione, il gioco risulta sicuramente più godibile e divertente.

Non è forse il miglior "pezzo" in gara per questa edizione dell'MSXdev ma è un buon prodotto.

di **Giampaolo Moraschi**







GIUDIZIO FINALE

» Giocabilità 75%
Le meccaniche di gioco sono particolari ma divertenti.

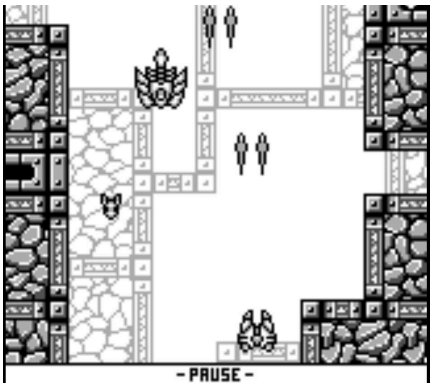
» Longevità 70%
In singolo annoia un po'. Trovatevi un amico!

NEW GAME

LUNAR LANCER

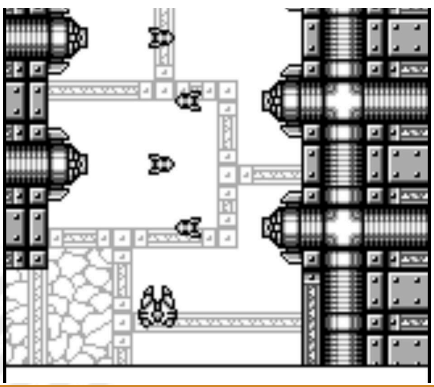
Anno: 2024
Editore/Sviluppatore: Jonas Fischbach
Genere: Shoot em up
Piattaforma: Game Boy
Sito web: https://jonas-fischbach.itch.io/lunar-lancer

È bello questo numero pieno di titoli per Game Boy e Game Boy color. La scena per questa console portatile che ha appena compiuto 35 anni è sempre molto attiva e molto viva.
Questo Lunar Lancer è un semplice e ben sviluppato shoot em up a scorrimento verticale, rilasciato dal suo autore come "regalo" di compleanno per Game Boy.



È un titolo molto lineare e semplice da apprendere: ci sono 5 lunghi livelli pieni di astronavi in caduta libera che vogliono farci la pelle, ci sono le classiche boss fight con 5 boss piuttosto coriacei, c'è la nostra astronave che può incrementare la sua potenza di fuoco raccogliendo dei power up lungo i livelli … insomma tutto classico nel suo genere a parte che i power up hanno un numero di munizioni limitate ed è presente… il Konami code! Il codice "trucco" presente in molti titoli della casa giapponese che diventa essenziale per poter proseguire nell'avventura. Lunar Lancer è un buon gioco che non porta innovazione nel genere ma che trova la sua "casa" proprio tra le mura della console portatile Nintendo. Sembra fatto apposta per essere giocato tra una fermata e l'altra del bus/metro/treno.
C'è anche un buon aspetto visivo che permette di distinguere correttamente tutto quello che si muove sullo schermo e ad una discreta velocità d'esecuzione. Ci sono anche alcuni livelli con fondale in parallasse piuttosto apprezzabili e musicalmente fa buon uso dei canali audio della console.
Lo sviluppatore ha inserito un sistema di password che permette di proseguire l'avventura da dove la si è lasciata e alcuni "easter eggs" molto particolari. Un gioco molto apprezzato dal sottoscritto che vi consiglia di darci un'occhiata e magari di caricarlo sul vostro portatile. Ne vale la pena.

di **Giampaolo Moraschi**



GIUDIZIO FINALE

» Giocabilità 85%
Un titolo semplice da affrontare e da apprendere con diversi spunti interessanti che "ampliano" il gameplay

» Longevità 80%
Senza il Konami code in alcuni punti diventa "ostico". La curva di difficoltà punta verso l'altro ma ci piace così.

NEW GAME

TIBURONCY

Anno: 2024
Editore/Sviluppatore: Patagonia
Genere: Action game
Piattaforma: Commodore 64
Sito web: https://patagonia.itch.io/tiburoncy-c64

Tiburoncy è un giochino semplice. Un mix tra un running game e il classico gioco d'azione dove aiutiamo un buffo squaletto con il salvagente nel salvataggio dei suoi amichetti intrappolati da pescatori.
Un titolo adatto per essere giocato assieme ad un pubblico più giovane grazie ad una grafica colorata, dinamiche e ambientazione.
Tiburoncy è realizzato dal collettivo Patagonia per il Flashparty 2024 in Argentina, un evento dedicato alla demoscene e ai videogiochi per tutti i computer in Sud America e nei paesi di lingua spagnola.
La scena ludica e demo nei paesi ispanici è piuttosto solida e sono molto validi gli sviluppatori. Se volete vedere qualcosa di più vi lascio qui il link all'evento:
https://flashparty.rebelion.digital/index.php/es/.
Tornando a Tiburoncy non si può pretendere un gioco dalla profondità estrema. Qui si salta in testa ai nemici che ci corrono incontro, si raccolgono chiavi, si aprono leve e si passa al livello successivo.
Agli occhi di un bambino/a piccolo sicuramente è divertente e coloratissimo con una bella grafica cartoonesca che si fa apprezzare.
La colonna sonora sente dell'esperienza del collettivo nella scena demo. Bella pompata, forse distaccata dall'azione, ma decisamente ben fatta.
Siamo di fronte ad un titolo semplice anche dal punto di vista dell'apprendimento. Con il tasto di fuoco facciamo nuotare Tiburoncy e premendo in basso colpiamo il nemico, schiacciandolo.
Non pensate ad ore ed ore di gioco. E' un titolo semplice e va bene così. Se avete un bambino piccolo è perfetto.

di **Giampaolo Moraschi**

GIUDIZIO FINALE

» Giocabilità 70%
Un tasto per nuotare e il sedere per schiacciare i nemici.

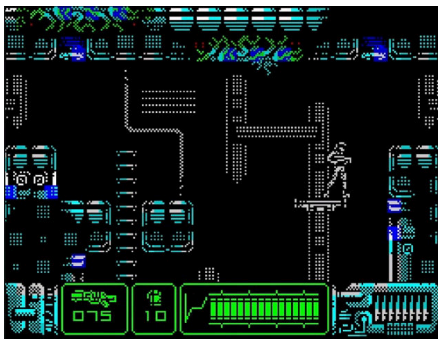
» Longevità 65%
Si porta facilmente a termine ma è divertente con i propri figlioli.

NEW GAME

ALIENS NEOPLASMA 2

Anno: 2024
Editore/Sviluppatore: Sinc LAIR
Genere: Action/Platform
Piattaforma: ZX Spectrum 128k, ZX Spectrum NEXT
Sito web: https://sinc-lair.itch.io/aliens-neoplasma-2

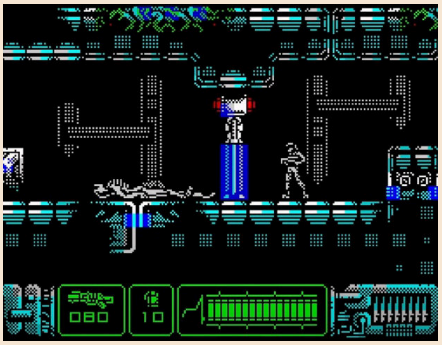
Il primo Alien Neoplasma era un grandissimo gioco, questo gioco è un dannato capolavoro!
Ambientato nell'universo di Aliens e seguito diretto del già citato primo capitolo vede la fuga del tenente Smith dal laboratorio segreto della Weyland-Yutani.

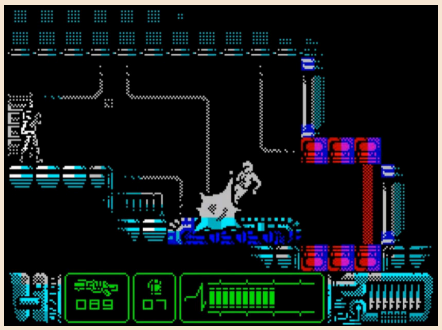


Inizialmente armati di un semplice cacciavite, la nostra eroina dovrà approcciare l'avventura in modo intelligente e scaltro per poter riuscire ad uscire viva dai lavoratori e scoprire cosa sta succedendo realmente.
Un prodotto realizzato con maestria in entrambe le versioni.
Quella per NEXT supporta la frequenza di aggiornamento schermo a 60 Hz e mostra un comparto tecnico di grande pregio con un uso dei colori magistrale.
La versione per ZX 128k è forse la vetta più alta per un titolo su questo home computer. Curato nelle animazioni e nell'aspetto e meraviglioso da ascoltare.
Il primo titolo brillava per giocabilità e questo secondo prodotto la perfeziona ancora e in meglio. Il sistema di controllo è meraviglioso e permette il controllo di ogni azione della protagonista. L'interazione con l'ambiente e l'obbligo di ragionare aumenta di tantissimo la curva della longevità.
In certi punti è ancora "pixel perfect" e occorre un po' di pratica, ma è senza dubbio uno dei migliori titoli moderni per ZX 128 e uno dei best games per NEXT in questo 2024.
Non lasciatevelo sfuggire.

Ultima cosa: la versione NEXT è instabile in emulazione.

di **Roberto Del Mar Pirazzini**





GIUDIZIO FINALE

» Giocabilità 95%
Immediato, dinamico e con una grande struttura di gioco. Bello dal primo all'ultimo minuto.

» Longevità 95%
Resterete attaccati allo schermo fino alla fine e se questo non vi basta ci sono ben tre livelli di difficoltà ulteriori.

NEW GAME

TWIST OF SOULS

Anno: 2024
Editore/Sviluppatore: LarsImNetz
Genere: Platform
Piattaforma: Atari 8bit
Sito web: https://forums.atariage.com/topic/374525-twist-of-souls/

Pixel Perfect! Decisamente Pixel Perfect questo Twist of Souls. Porting del titolo di Fran Tovar visto su PC nel 2018, si tratta di un platform game esplorativo dove guideremo una pulzella all'interno di un tempio antico e misterioso che custodisce una maledizione.

La meccanica di gioco è interessante. Per poter affrontare il gioco è necessario "sacrificarsi" e ribaltare la visione classica. In poche parole, in certi momenti sarà possibile affrontare il livello capovolgendolo. Questo avviene tramite la pressione della barra spaziatrice una volta attraversata una grande porta rossa. Il mondo capovolto permette di affrontare i livelli in modo completamente diverso e di raggiungere posti altrimenti impossibili nel "metodo canonico".
Il titolo è arrivato terzo al recente ABBUC contest 2024 e richiede un Atari con almeno 64Kb di Ram.
La trasposizione tecnica è praticamente perfetta e si adatta alle caratteristiche grafiche della macchina. Non che fosse complicato da riportare vista la componente minimalista del titolo per PC.

Sonoro minimalista che accompagna tutti noi giocatori in un inferno fatto di salti quasi impossibili e di una buona dose di frustrazione.
Il gioco è ben sviluppato ma è decisamente difficile anche per il più "sgamato" dei giocatori.
Se vi piacciono le sfide impossibili è il titolo che fa per voi!

di **Giampaolo Moraschi**

GIUDIZIO FINALE

» Giocabilità 75%
Meccanica di gioco interessante.

» Longevità 75%
Difficile e pesantemente Pixel Perfect.

NEW GAME

CHIMNEY MAN

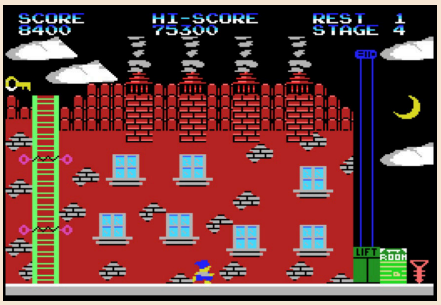
Anno: 2024
Editore/Sviluppatore: Gamecast
Genere: Action/Platform
Piattaforma: ZMSX, 32KB
Sito web: https://www.msxdev.org/wp-content/uploads/2024/06/MSXdev24_ChimneyMan_v1.5.zip

Abbiamo ricevuto una gentile richiesta da parte di **GameCast** di recensire il loro gioco CHIMNEY MAN per MSX, sviluppato per partecipare alla MSXDEV24.
Ovviamente non ce lo siamo fatti ripetere due volte. :-)

Chimney Man é un simpatico platform game dove vestirete i panni di uno spazzacamino che deve recuperare i suoi strumenti di lavoro per pulire i camini di tutti i palazzi della cittá.
Purtroppo gli arnesi somo chiusi in uno sgabuzzino chiuso a chiave e la chiave si trova in cima ad un palazzo.
Va beh, cosa vuoi che sia, siamo spazzacamini ed arrampicarci sui palazzi é il nostro mestiere. Peccato peró che la scala per raggiungere il tetto sia attraversata da letali scariche elettriche...
Armatevi quindi di tempismo e pazienza ed andate a recuperare la chiave!
Il concept é decisamenete carino, ma... é tutto qui. Dopo aver completato il primo ed il secondo livello, che differiscono soltanto per il numero di camini, 2 nel primo e 4 nel secondo, gli altri livelli si ripetono allo stesso modo. Peccato, perché la grafica é molto carina, colorata al punto giusto e piacevole a vedersi, in pieno stile retro d'antan. Anche il sonoro non é male.
Probabilmente i tempi stretti della MSXDEV non hanno concesso la possibilitá di aggiungere altre feature al programma; cosa che invece noi consigliamo caldamente di fare.
La base per un bel gioco c'é tutta, va soltanto aggiunto un po' di varietá per ottenere un titolo che potrebbe a tutti gli effetti diventare un simpatico prodotto commerciale.

Lo potete giocare anche online: https://www.file-hunter.com/MSXdev/?id=chimneyman.

di **Francesco Fiorentini**



GIUDIZIO FINALE

» Giocabilità 70%
Ci vuole poco per capire cosa fare ed altrettanto per completare i livelli.

» Longevità 50%
Ma dopo aver scoperto che ci sono solo 2 livelli, sará difficile che ci rigiocherete.

BACK TO THE FUTURE II

Anno: 1990

Editore/Sviluppatore: Imageworks

Genere: Tie-in

Piattaforma: Commodore 64

Ricordo come fosse ieri quella vigilia di Natale in cui mio padre mi portò a casa un doppio regalo, ossia Golden Axe! Di corsa mi recai in negozio con lo scontrino per farmelo cambiare e così presi il gioco di Ritorno al futuro 2: sugli schermi del Commodore 64 e di tante altre piattaforme con il titolo originale di Back to the future part II della Images Software. Sono sempre stato un po' scettico sui tie in, ma con un titolo come questo che comprendeva diversi livelli misti con la giocabilità diversa tra loro beh, il bello per me è stato proprio questo, senza contare la musica originale che ci accompagnava per tutto il caricamento facendoci rivivere l'atmosfera natalizia degli anni 90 tra prime visioni e giochi! Cominciamo dal primo livello, in cui noi impersoneremo Marty McFly a bordo di un volo pattino nell'anno 2015 (qualcuno ne ha visto uno?), in cui dovremo raggiungere il municipio su strade ed incroci con scorrimento orizzontale e verticale. Disturbati da nemici, anch'essi in volo pattino, da un Biff vecchiotto con il suo bastone che ricorda il portinaio dei cortili e da vari ostacoli come, per esempio, le pozzanghere. Verranno in nostro aiuto vari bonus, come energia aggiuntiva e incentivi vari. Nel secondo livello ci vede all'interno di un abitazione con visuale dall'alto, dove dovremo fare in modo di far uscire Jennifer senza farla incontrare con gli altri occupanti; tutto ciò richiede un minimo di strategia ma nulla di impossibile. Il terzo livello invece é un picchia-duro a scorrimento orizzontale sulla Hill Valley. Ormai in preda dei malviventi scagnozzi di Biff, a suon di pugni e armi si dovremo raggiungere l'altra sponda del livello dove c'é il Doc con la sua De Loren ad aspettarci. Il quarto livello è un puzzle a tutti gli effetti, scopo del gioco? Lo sanno tutti, ma se lo trovate tosto non preoccupatevi, una volta terminato il tempo sarà possibile ricominciare da lí, idem per tutti gli altri. Il quinto e ultimo livello é identico al primo, solo che ora avremo nemici anche a piedi, un macchinone nero guidato da Biff che cercherà di stirarci. Alla fine la sequenza animata più divertente del film ed in più il diario nelle nostre mani. La grafica e il sonoro li eleggerei come i migliori dell'anno e persino la giocabilità merita il tappeto rosso dato che la voglia di rigiocarlo non era poca. La longevità é dettata dal quarto livello con quel maledetto puzzle che ci fa sudare il cervello. C'e' da dire che nel caricamento, oltre alla leggendaria colonna sonora del titolo, i tempi di attesa non sono lunghi e all'inizio si puó assistera alla scena in cui la De Loren in tutto il suo splendore decolla a tutta velocità. Molti di voi, o almeno quelli che hanno visto il film, sapranno già che il finale potrebbe essere deludente... E invece è semplicemente un chiaro invito ad accaparrarsi tutta la trilogia sia in film che in videogame per rivivere quegli anni d'oro.

Con il Natale ormai alle porte, le luminarie già attive in alcuni comuni e la lista dei regali quasi al completo, possiamo goderci questo numero natalizio sotto l'albero e, senza essere troppo ripetitivo, tornare a giocare a questo ed altri titoli!

di **Daniele Brahimi**

GIUDIZIO FINALE

» Giocabilità 80%
Cinque livelli diversi tra loro, che si giocano molto bene.

» Longevità 80%
In alcuni tratti semplice, ma la voglia di rigiocarlo anche solo per ascoltare la musica...

[... continua dalla seconda pagina]

Conservazione di software e linguaggi: molti programmi e giochi venivano distribuiti con listati e codici sorgenti pubblicati su riviste o manuali. La loro digitalizzazione è essenziale per preservare il software d'epoca e le pratiche di programmazione di quegli anni.

Memoria culturale: le riviste degli anni 70, 80 e 90 offrono una panoramica unica sugli stili di vita e le aspirazioni dell'epoca, specialmente in relazione all'evoluzione dell'informatica e del gaming.

Supporto all'emulazione: le scansioni contengono documentazione utile per gli sviluppatori di emulatori, permettendo una maggiore precisione nella riproduzione delle console e dei computer vintage.

Recupero di contenuti perduti: molti giochi e applicazioni del passato non hanno mai avuto una distribuzione digitale e, senza le scansioni di manuali e riviste che ne parlano, rischierebbero di essere dimenticati o non ricostruibili.

Salvataggio dei ricordi di una generazione: per molti appassionati le scansioni digitali rappresentano una porta d'accesso alla propria infanzia o adolescenza, sono materiali che alimentano ricordi e nostalgia, aspetti significativi del retrogaming.

Risorsa educativa e didattica: i materiali digitalizzati possono essere usati per corsi di storia dell'informatica, cultura digitale e anche per insegnare le basi della programmazione attraverso esempi storici, rendendo più accattivante l'apprendimento.

Prevenire il rischio della perdita fisica: la carta, col tempo, si deteriora. La digitalizzazione garantisce che queste informazioni siano preservate oltre la vita fisica dei documenti originali, proteggendole da distruzione accidentale o deterioramento.

Per tutte queste ragioni la scansione digitale è un'attività estremamente preziosa per salvaguardare la memoria tecnologica e culturale legata ai computer e alle console del passato, rendendo disponibile un patrimonio altrimenti a rischio di scomparsa. Quindi invito anche voi a fare mente locale e, quando impugnate il vostro tablet a tarda sera per accingervi a leggere un manuale, un libro o una rivista del passato, pensate a come sdebitarvi nei confronti di tutti quei gruppi che impiegano tempo e risorse per combattere la battaglia del tempo per donarci pezzi di storia dei nostri amati retrocomputer. Soprattutto quelli che lo fanno in maniera totalmente disinteressata e con l'unico scopo della condivisione e della diffusione universale.

David La Monaca

Disclaimer

RetroMagazine World (fanzine aperiodica) è un progetto interamente no profit e fuori da qualsiasi circuito commerciale. Tutto il materiale contenuto è prodotto dai rispettivi autori e pubblicato grazie alla loro autorizzazione.

RetroMagazine World viene concessa al pubblico con licenza: Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale (CC BY-NC-SA 4.0 INT) https://creativecommons.org/licenses/by-nc-sa/4.0/deed.it

In pratica sei libero di: condividere, riprodurre, distribuire, comunicare o esporre in pubblico, rappresentare, eseguire e recitare questo materiale con qualsiasi mezzo e formato, modificare, rielaborare, trasformare il contenuto e basarti su di esso per altre opere, alle seguenti condizioni:

Attribuzione
Devi riconoscere una menzione di paternità adeguata, fornire un link alla licenza e indicare se sono state effettuate delle modifiche. Puoi farlo in qualsiasi maniera ragionevole possibile, ma non con modalità tali da suggerire che il licenziante avalli te o l'utilizzo del materiale da parte tua.

NonCommerciale
Non puoi utilizzare il materiale per scopi commerciali.

StessaLicenza
Se rielabori, trasformi il materiale o ti basi su di esso, devi distribuire i tuoi contributi con la stessa licenza del materiale originario.

Il licenziante non può revocare questi diritti fintanto che tu rispetti i termini della licenza.

Divieto di restrizioni aggiuntive
Non puoi applicare termini legali o misure tecnologiche che impongano ad altri soggetti dei vincoli giuridici su quanto la licenza consente loro di fare.

AI Disclaimer - Nessun contenuto di questa rivista digitale (editoriali, articoli, recensioni, testi o immagini) è stato prodotto o generato da strumenti AI.

RetroMagazine World
Anno 8 - Numero 49 - NOVEMBRE 2024

Direttore Responsabile
Francesco Fiorentini
Vice Direttore
Marco Pistorio
Coordinatore Redazione/Editing
David La Monaca
Web Manager/Social Media Manager
Giorgio Balestrieri/Carlo Nith Del Mar Pirazzini